

MedCalc-Eval and MedCalc-Env: Advancing Medical Calculation Capabilities of Large Language Models

Kangkun Mao¹, Jinru Ding¹, Jiayuan Chen¹, Mouxiao Bian¹, Ruiyao Chen¹, Xinwei Peng¹,
Sijie Ren¹, Linyang Li¹, Jie Xu^{1†}

¹*Shanghai AI Laboratory*; [†]*Correspond to: {xujie}@pjlab.org.cn*

Abstract

As large language models (LLMs) become increasingly integrated into the medical domain, existing benchmarks have primarily focused on evaluating their capabilities in question answering and descriptive reasoning. However, real-world clinical practice often relies on quantitative reasoning through medical calculators based on equations and rule-based scoring systems, which are essential for evidence-based decision-making. Current benchmarks such as MedCalc-Bench cover only a limited number of calculation tasks and do not comprehensively reflect LLMs's performance in practical clinical computation scenarios.

To address this gap, we introduce MedCalc-Eval, the largest and most comprehensive benchmark for evaluating LLMs' capabilities in medical calculations. MedCalc-Eval includes over 700+ distinct clinical calculation tasks, categorized into two major types: equation-based calculations (e.g., Cockcroft-Gault, BMI, BSA) and rule-based scoring systems (e.g., Apgar Score, CHA2DS2-VASc, Glasgow Coma Scale). These tasks span a wide range of clinical specialties, including internal medicine, surgery, pediatrics, critical care, obstetrics and gynecology, emergency medicine, neurology, cardiology, pulmonology, urology, and more, creating a significantly broader and more challenging evaluation setting compared to existing benchmarks.

To further enhance LLM performance in medical computation, we also present MedCalc-Env, a reinforcement learning environment built on the InternBootcamp framework. MedCalc-Env is specifically designed to train LLMs in multi-step clinical reasoning and action planning within interactive settings. Using this environment, we fine-tuned a Qwen2.5-32B model with reinforcement learning, achieving state-of-the-art (SOTA) performance on MedCalc-Eval.

Our evaluation demonstrates that the RL-trained model exhibits markedly improved numerical sensitivity, formula selection accuracy, and reasoning robustness across a wide range of medical calculation tasks. Nonetheless, challenges remain in areas such as unit conversion, multi-condition logic, and context understanding. We hope our work sheds light on the quantitative reasoning gaps in current LLMs and inspires further advancements toward building reliable clinical decision support systems powered by AI. Implementation details with related code and datasets will be updated at <https://github.com/maokangkun/MedCalc-Eval>.¹

1 Introduction

The rapid advancements in large language models (LLMs) have led to their increasing integration into various specialized domains, including medicine. These models have demonstrated remarkable capabilities in tasks such as medical question answering, information extraction, and descriptive reasoning. However, the existing benchmarks predominantly focus on these qualitative aspects,

¹This project is ongoing. We welcome feedback from the community and will frequently update our work.

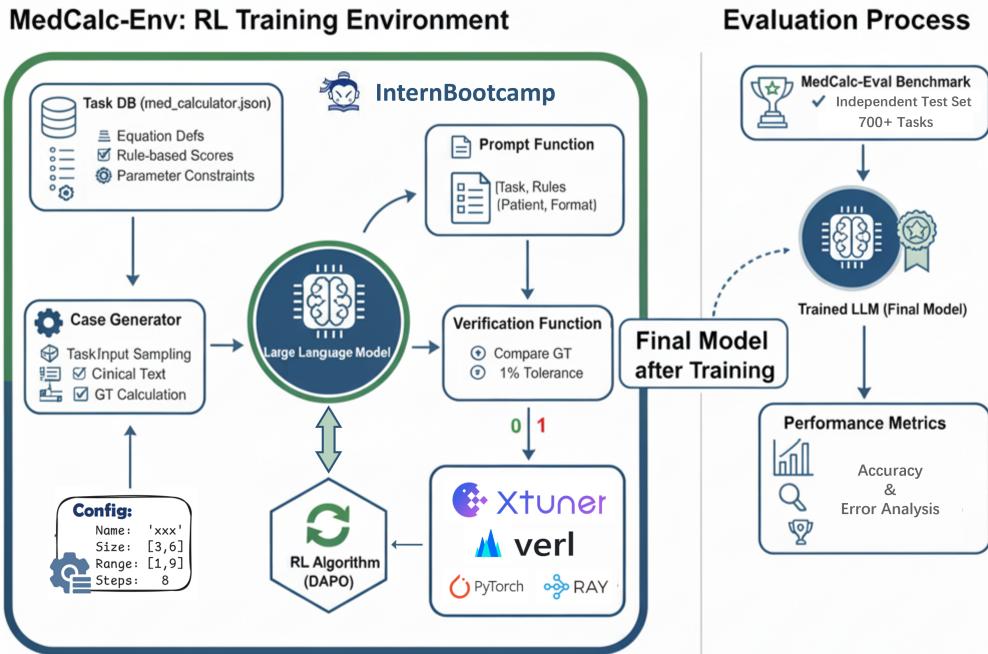


Figure 1: Overview of the MedCalc-Env training framework and the MedCalc-Eval evaluation process. The left side illustrates the reinforcement learning-based MedCalc-Env training loop: (1) The Case Generator samples from the task database to create clinical calculation cases with ground truth answers; (2) The Prompt Function formats the case into an input for the LLM; (3) The LLM generates reasoning steps and a final answer; (4) The Verification Function compares the LLM’s answer with the ground truth to generate a reward signal; and (5) The RL algorithm uses this reward signal to update the LLM’s model weights. This cycle repeats continuously to enhance the model’s capabilities. The right side shows the evaluation process: after full training, the final model is tested on the independent and comprehensive MedCalc-Eval benchmark to objectively measure its final performance and generalization ability on medical calculation tasks.

often overlooking the critical need for quantitative reasoning in real-world clinical practice. Medical professionals frequently rely on clinical calculators, which are built upon precise equations and rule-based scoring systems, to make evidence-based decisions. These tools are indispensable for accurate diagnosis, prognosis, and treatment planning.

Despite the importance of quantitative reasoning, current evaluation benchmarks for LLMs in medicine, such as MedCalc-Bench (Khandekar et al., 2023), offer a limited scope. While MedCalc-Bench was a pioneering effort to introduce clinical calculation scenarios, it covers only a restricted number of calculation tasks and does not fully capture the complexities and breadth of practical clinical computation. This gap highlights a significant challenge: without robust quantitative reasoning capabilities, LLMs cannot fully support the intricate decision-making processes inherent in medical practice.

To address this critical limitation, we introduce **MedCalc-Eval**, a novel and comprehensive benchmark designed specifically for evaluating the medical calculation capabilities of LLMs. MedCalc-Eval significantly expands upon previous efforts by encompassing over 700 distinct clinical calculation tasks. These tasks are meticulously categorized into two primary types: equation-based calculations, which involve direct mathematical formulas (e.g., Cockcroft-Gault formula for creatinine clearance, Body Mass Index (BMI), Body Surface Area (BSA)), and rule-based scoring systems, which require logical inference based on predefined criteria (e.g., Apgar Score for newborn assessment, CHA2DS2-

VASc score for stroke risk in atrial fibrillation, Glasgow Coma Scale for assessing consciousness). The benchmark spans a wide array of clinical specialties, including internal medicine, surgery, pediatrics, critical care, obstetrics and gynecology, emergency medicine, neurology, cardiology, pulmonology, urology, and many others. This broad coverage creates a significantly broader and more challenging evaluation setting compared to existing benchmarks, providing a more accurate assessment of LLMs's performance in diverse clinical computation scenarios.

Furthermore, to facilitate the enhancement of LLM performance in medical computation, we also present **MedCalc-Env**, a reinforcement learning environment. Built upon the robust InternBootcamp framework, MedCalc-Env is specifically engineered to train LLMs in multi-step clinical reasoning and action planning within interactive settings. This environment allows models to learn from iterative interactions and receive immediate feedback, thereby refining their ability to handle complex medical calculation tasks. Through the utilization of MedCalc-Env, we successfully fine-tuned a Qwen2.5-32B model using reinforcement learning techniques. This RL-trained model achieved state-of-the-art (SOTA) performance on both our newly introduced MedCalc-Eval benchmark and the existing MedCalc-Bench, demonstrating the effectiveness of our approach in improving quantitative reasoning in LLMs.

Our comprehensive evaluation reveals that the reinforcement learning-trained model exhibits significantly improved numerical sensitivity, enhanced formula selection accuracy, and greater reasoning robustness across a broad spectrum of medical calculation tasks. However, our analysis also identifies persistent challenges, particularly in areas such as unit conversion, multi-condition logic, and nuanced context understanding. These findings underscore the remaining quantitative reasoning gaps in current LLMs and highlight crucial areas for future research. We believe that our work not only provides a more rigorous evaluation framework but also inspires further advancements towards the development of highly reliable and intelligent clinical decision support systems powered by AI. (Figure 1)

Our contributions can be summarized as follows:

- We introduce **MedCalc-Eval**, the largest and most comprehensive benchmark for evaluating LLMs's medical calculation capabilities, covering over 700+ tasks across various specialties.
- We develop **MedCalc-Env**, a reinforcement learning environment based on InternBootcamp, designed to train LLMs in multi-step clinical reasoning for medical computations.
- We demonstrate that an RL-trained Qwen2.5-32B model achieves state-of-the-art performance on both MedCalc-Eval and MedCalc-Bench, showcasing the effectiveness of our proposed training methodology.
- We provide a detailed error analysis, identifying key challenges and future research directions for improving quantitative reasoning in medical LLMs.

2 MedCalc-Eval: A Comprehensive Evaluation Benchmark

To overcome the limitations of existing benchmarks and provide a more rigorous evaluation of LLMs's medical calculation capabilities, we propose **MedCalc-Eval**. This benchmark is designed to be the largest and most comprehensive of its kind, encompassing a vast array of clinical calculation tasks that closely mirror real-world medical practice. MedCalc-Eval significantly expands the scope and depth of evaluation, offering a more accurate assessment of LLMs's performance in diverse and challenging clinical scenarios.

2.1 Task Definition

Medical clinical calculator tasks, in the context of clinical diagnosis and treatment, involve a quantitative reasoning process based on patient medical record information and established medical formulas or scoring rules. These tasks typically comprise three critical stages:

1. **Knowledge Recall:** Correctly identifying and invoking the appropriate clinical formula or scoring scale relevant to the given medical scenario.
2. **Information Extraction:** Accurately extracting relevant parameters (numerical values, categories, time points, etc.) from lengthy or complex medical record texts. This often requires sophisticated natural language understanding to identify key data points amidst noise.
3. **Numerical Reasoning:** Performing multi-step calculations and logical judgments based on correctly substituted parameters, and outputting the final result. This stage demands precision, adherence to specific rules, and often involves complex arithmetic or conditional logic.

Unlike traditional open-domain question answering or descriptive reasoning tasks, medical clinical calculator tasks are highly structured, emphasizing precision, compliance, and verifiability. They form a crucial component of Clinical Decision Support Systems (CDSS) and serve as a vital indicator for assessing the practical applicability of LLMs in medical settings.

2.2 Construction of MedCalc-Eval

MedCalc-Eval is meticulously constructed to provide a comprehensive and challenging evaluation environment. As summarized in Table 2 (see Appendix A), the benchmark contains a total of **709 distinct clinical calculation tasks** (629 formula-based and 80 scale-based), making it the largest collection of its kind. Its key characteristics include:

- **Task Scale and Diversity:** MedCalc-Eval includes **132 formula-based categories** and **27 scale-based categories**, covering a wide range of clinical scenarios. This extensive scale ensures a broad coverage of medical calculation scenarios.
- **Task Types:** The benchmark categorizes tasks into two major types:
 - **Equation-based calculations:** These involve direct application of mathematical formulas, such as the Cockcroft-Gault formula for creatinine clearance, Body Mass Index (BMI), and Body Surface Area (BSA).
 - **Rule-based scoring systems:** These require logical inference and scoring based on predefined criteria, exemplified by the Apgar Score, CHA2DS2-VASc score, and Glasgow Coma Scale.
- **Specialty Coverage:** MedCalc-Eval spans dozens of clinical specialties. As shown in Figure 2 and detailed in Table 3 (see Appendix A), the top formula-based categories include Laboratory Medicine, Pulmonary Diseases, and Nephrology, while scale-based questions frequently appear in Cardiovascular Diseases and Obstetrics and Gynecology. This broad specialty coverage ensures that the benchmark reflects the diverse needs of real-world clinical practice.
- **Increased Difficulty:** To provide a more realistic and challenging evaluation, MedCalc-Eval incorporates scenarios that involve multi-condition judgments, complex formula nesting, and cross-unit conversions. The most frequently required input indicators, such as **Weight**, **Age**, and **Height**, often require careful unit handling, as detailed in Table 4 (see Appendix A).

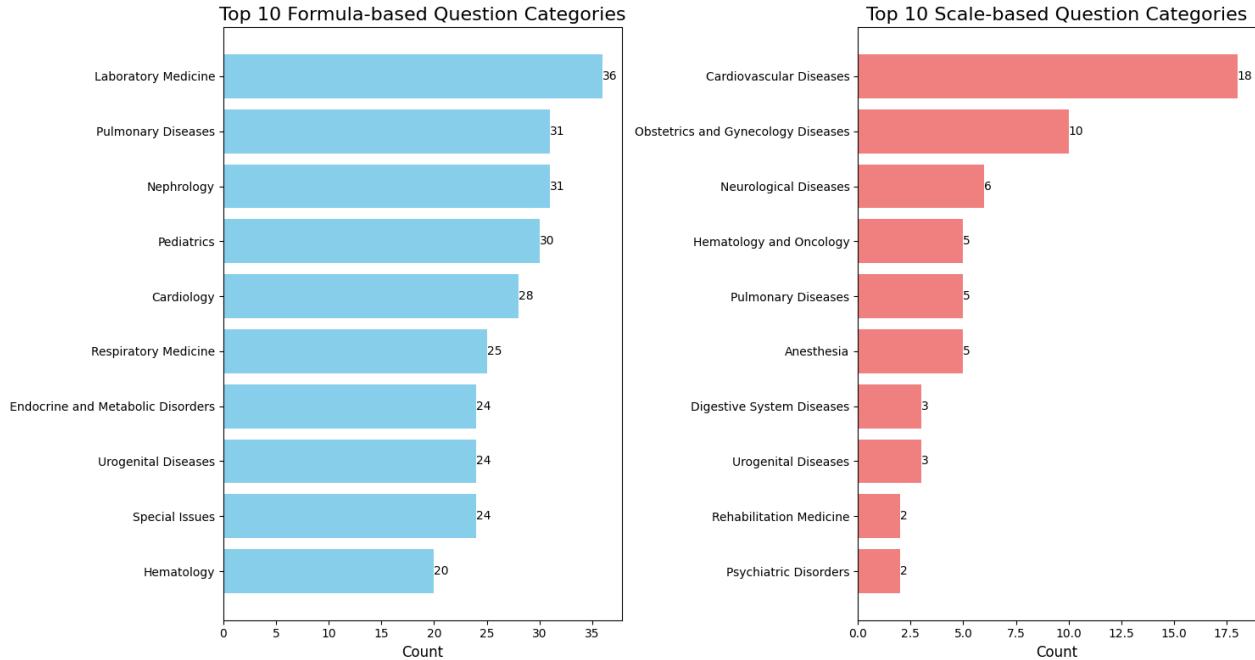


Figure 2: Top 10 categories in MedCalc-Eval

2.3 Comparison with MedCalc-Bench

While MedCalc-Bench (Khandekar et al., 2023) laid the groundwork for evaluating LLMs in medical calculation, MedCalc-Eval represents a significant advancement in both breadth and depth. MedCalc-Bench covered 55 calculators and 1047 instances, with specific error tolerances for different task types. However, its limitations included a relatively small number of calculation tasks, insufficient specialty coverage, and a lack of challenging scenarios involving complex reasoning chains, multi-modal inputs, or cross-unit conversions. These limitations suggest that while MedCalc-Bench laid a foundational framework, it does not fully characterize the true capabilities and boundaries of LLMs in complex medical calculation tasks.

MedCalc-Eval directly addresses these shortcomings by:

- **Vastly expanded task variety:** Offering **709 distinct tasks** compared to MedCalc-Bench's 55, providing a much richer and more diverse evaluation landscape.
- **Comprehensive specialty representation:** Covering a wider range of clinical specialties, ensuring that LLMs are tested across the full spectrum of medical domains.
- **Enhanced complexity:** Introducing more challenging scenarios that demand advanced reasoning, such as nested formulas, conditional logic, and unit conversions, which are crucial for practical clinical application.

In summary, MedCalc-Eval significantly extends the evaluation system's breadth and depth, providing a more stringent and systematic test for the medical calculation capabilities of LLMs. It serves as a more robust platform for future research and development in this critical area.

3 MedCalc-Env: A Reinforcement Learning Environment

Beyond the comprehensive evaluation benchmark, we have also developed **MedCalc-Env**, a novel reinforcement learning (RL) environment built upon the InternBootcamp framework. MedCalc-Env is specifically designed to train large language models in multi-step clinical reasoning and action planning within interactive settings, thereby enhancing their quantitative reasoning capabilities in medical computation tasks. The overall architecture of MedCalc-Env is illustrated in Appendix D.

3.1 Case Generator

The case generator in MedCalc-Env is crucial for producing diverse and realistic clinical scenarios, ensuring both input randomness and the verifiability of output results. The overall logic of the case generator is illustrated in the following. It operates by sampling tasks from two main categories: equation (formula-based) and scale (rule-based), using the detailed configuration JSON file.

3.1.1 Overall Process

The process begins with **Task Sampling**, where a category (either equation or scale) is randomly selected, followed by the random selection of a specific calculator or scoring table within that category. Subsequently, the `gen_a_case` method is invoked to generate a complete instance based on the formula definitions, indicator constraints, and scale items defined in the configuration file (e.g., `med_calculator.json`).

3.1.2 Formula-based Tasks

For formula-based tasks, the system performs **Input Sampling** by iterating through the required input parameters for a given formula. It reads the type and range from the indicator dictionary in the configuration file:

- **Integer (int)**: A random integer is selected within the specified range.
- **Float (float)**: A random float value is generated within the range, with precision maintained as per constraints.
- **Choice (choice)**: A random item is chosen from a given set of options (list or dictionary), returning the mapped numerical value if it's a dictionary.

These sampled parameters are then transformed into **Clinical Expression**, concatenating parameter values with units to form patient information descriptions (e.g., "blood pressure 120 mmHg"). The sampled results are then substituted into the original formula for **Formula Substitution and Solving**. The system uses Python's math library and eval function to compute the result. Error handling is implemented to re-sample if invalid values (e.g., division by zero) occur. Finally, **Output Standardization** ensures the target value is rounded as per the result indicator's configuration.

3.1.3 Scale-based Tasks

For scale-based tasks, the system iterates through all items in the scale. For **Single-choice** items, one option is randomly selected, and its corresponding score is accumulated. For **Multi-choice** items, a random number of options are selected, and their scores are summed. These selections are then used to generate **Input Descriptions** that mimic real medical records (e.g., "symptoms: chest pain, shortness of breath..."). The **Score Calculation** sums the scores of selected options to derive the final target score.

3.1.4 Randomization and Robustness

MedCalc-Env incorporates several mechanisms to ensure the diversity and robustness of generated cases:

- **Multi-source Randomness:** Including task categories, specific calculators, input values/options, and the inclusion of rule descriptions, ensuring diverse data distribution.
- **Error Tolerance:** Catching and re-sampling for errors like division by zero or invalid values, ensuring all generated results are valid.
- **Enhanced Realism:** Automatically concatenating units and labels in input descriptions to conform to clinical medical record conventions.
- **Random Inclusion of Rules:** Each case has a probability (controlled by the `add_rule_ratio` parameter) to include the formula explanation or scale scoring criteria in the prompt. This feature allows for evaluating the model's performance in both scenarios: when it must rely solely on its internal medical knowledge versus when explicit calculation rules are provided.

3.2 Prompt Function Setting

The prompt function serves as the "interface contract" connecting the large language model with the verification environment. It structures the task intent, input data, and constraints into executable instructions, and constrains the model's output to a machine-readable and verifiable format. Two main variations of prompt templates are designed based on whether knowledge and rules are provided. The detailed prompt templates are provided in Appendix B.

3.3 Reinforcement Learning with Verifiable Rewards (RLVR)

For our training paradigm, we use **Reinforcement Learning with Verifiable Rewards (RLVR)** (Wen et al., 2025), a method that implicitly incentivizes correct reasoning in the base LLM by providing a reward signal based on the verifiable final output. The core idea is to leverage the deterministic nature of medical calculation tasks to create a strong, objective reward signal.

The verification function determines the reward signal:

- **Reward = 1:** If the model's final answer, extracted from the boxed output, matches the ground truth value calculated by the simulation tool.
- **Reward = 0:** Otherwise.

To account for potential minor numerical discrepancies inherent in complex, multi-step calculations and floating-point arithmetic, we introduce an error tolerance for formula-based tasks (e.g., laboratory/physical/dosage conversions). Specifically, an error tolerance of $\pm 1\%$ is allowed, meaning if the difference between the model's predicted result and the ground truth falls within this range, it is still considered correct. This approach facilitates more robust training by focusing on the correctness of the final numerical result, which is the ultimate verifiable metric in these tasks. This verifiable reward mechanism, by focusing the reward signal solely on the final verifiable output, implicitly incentivizes the model to generate correct, multi-step reasoning, thereby addressing extraction and calculation errors.

4 Experiments

This section details the experimental setup, presents the results obtained from evaluating our RL-trained model on MedCalc-Eval and MedCalc-Bench, and provides an in-depth analysis of the error types observed.

4.1 Experimental Setup

Our experimental validation involved leveraging the MedCalc-Env to train models using the Reinforcement Learning with Verifiable Rewards (RLVR) methodology (Wen et al., 2025), demonstrating its effectiveness in enhancing quantitative reasoning.

4.1.1 Model Selection & Training Workflow

We employed the **Qwen2.5-7B** and **Qwen2.5-32B** (Qwen et al., 2025) model for our Reinforcement Learning with Verifiable Rewards (RLVR) experiments, which was fine-tuned within the MedCalc-Env environment. We also evaluated several other large language models (Qwen et al., 2025; Yang et al., 2025; DeepSeek-AI et al., 2025) in a zero-shot setting for comparison. As for training process, we total generated 10k corresponding data instances for all formula and scale in the MedCalc-Env bootcamp to construct the training environment for RLVR. During the RLVR stage, the following key training parameters were configured:

- **Algorithm:** Dynamic sAmpling Policy Optimization (DAPO) (Yu et al., 2025), with No KL-loss.
- **Training Prompt Batch Size:** 128.
- **Maximum Response Length:** 8192.
- **Temperature:** 1.
- **Rollout Number:** 8.
- **Loss Aggregation Mode:** token-mean.

We utilized **xpuyu-RL** (Contributors, 2023) and **VeRL** (Sheng et al., 2025) frameworks for the training process.

4.1.2 Experimental Data

- **Training Data:** The training dataset was exclusively generated using the MedCalc-Env bootcamp, ensuring consistency and relevance to the medical calculation tasks.
- **Testing Data:** For evaluation, we used a separate test set generated by the MedCalc-Env bootcamp and subsequently verified manually, ensuring high quality and reliability of the evaluation.

4.2 Experimental Results and Analysis

Our experiments yielded significant improvements in the LLMs's medical calculation capabilities, demonstrating the effectiveness of the proposed MedCalc-Env training environment.

4.2.1 Main Results

Figure 3 illustrates the performance comparison of various large language models on both the MedCalc-Eval and MedCalc-Bench benchmarks. The detailed data is provided in Appendix 5.

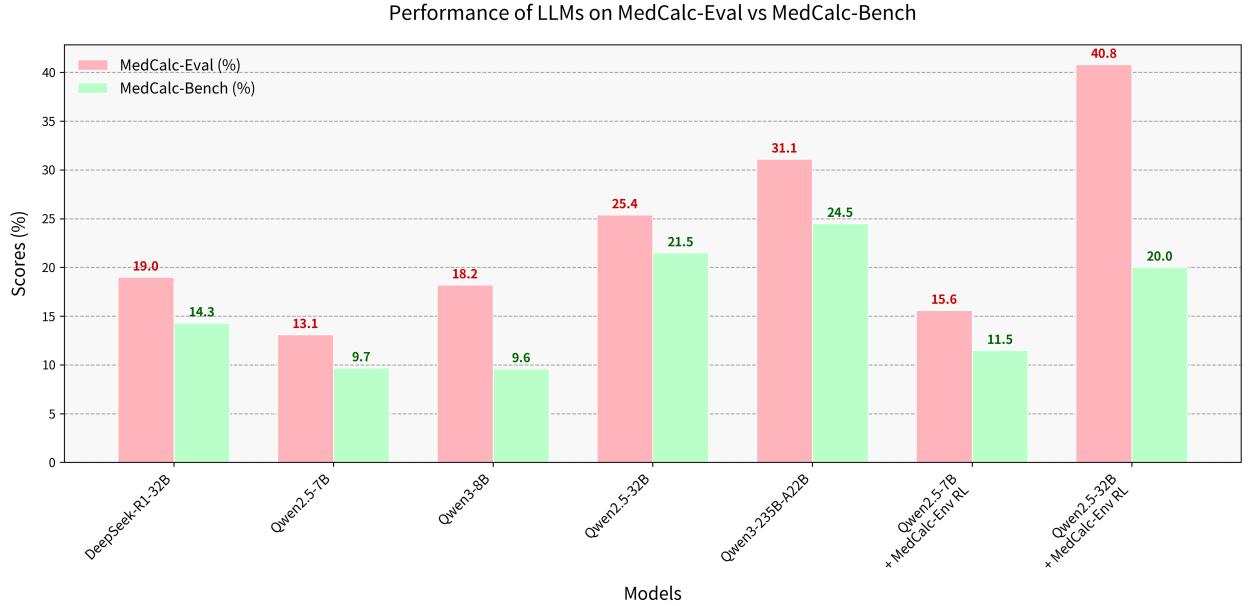


Figure 3: Performance of LLMs on MedCalc-Eval compared to MedCalc-Bench

The results show that:

- The performance of all evaluated large language models is generally poor on both **MedCalc-Eval** and **MedCalc-Bench**. The accuracy of most models is below 30%, with the best zero-shot performance from **Qwen3-235B-A22B** only reaching 31.1% on MedCalc-Eval. This collectively demonstrates that existing LLMs still have significant shortcomings in medical calculation tasks, indicating a large room for improvement. Furthermore, MedCalc-Eval (Chinese medical calculation) and MedCalc-Bench (English medical calculation) serve as complementary benchmarks for comprehensively evaluating the medical calculation capabilities of LLMs.
- The Reinforcement Learning (RL) approach trained within our **MedCalc-Env** environment leads to substantial performance gains. Specifically, the **Qwen2.5-32B + MedCalc-Env RL** model achieved the highest accuracy of 40.8% on MedCalc-Eval, representing an absolute improvement of 15.4% over its base model (25.4%). This substantial gain underscores the effectiveness of our proposed RL training environment in enhancing quantitative reasoning for medical tasks.
- The largest model, **Qwen3-235B-A22B**, shows the highest zero-shot performance (31.1%), but the RL-trained Qwen2.5-32B model surpasses it by a large margin (40.8%), indicating that specialized training with MedCalc-Env is more effective than simply scaling up model size.

4.2.2 Ablation Study

To further validate the effectiveness of the MedCalc-Env training, we conducted an ablation study using the Qwen2.5-7B model, as summarized in Appendix 6.

The ablation study confirms the positive impact of the RL training:

Table 1: Ablation study results of Qwen2.5-7B model

Model	MedCalc-Eval (%)	MedCalc-Bench (%)	AIME24 (%)
Qwen2.5-7B	13.1	9.7	16.7
Qwen2.5-7B + MedCalc-Env RL	15.6	11.5	20.0

- The **Qwen2.5-7B + MedCalc-Env RL** model shows a consistent improvement across all three evaluation sets: MedCalc-Eval (13.1% to 15.6%), MedCalc-Bench (9.7% to 11.5%), and the general mathematical evaluation AIME24 (16.7% to 20.0%).
- Notably, the model trained primarily with Chinese medical data in the MedCalc-Env environment also achieved performance improvement on the English medical calculation benchmark MedCalc-Bench (from 9.7% to 11.5%), demonstrating that the performance gain stems from an enhancement in the model’s fundamental medical calculation capabilities, which are largely language-agnostic.
- The improvement on AIME24 (an increase of 3.3% absolute, or 20% relative) is particularly noteworthy, as it was trained with a mixed dataset comprising 10,000 medical data instances and 17,000 mathematical data instances (DAPO_MATH_17K). This result validates that integrating specialized medical knowledge through our RL environment does not lead to a loss of generalization performance in other domains, but rather **enhances the model’s core quantitative reasoning capabilities**.

4.2.3 Case Study & Error Analysis

To gain deeper insights into the model’s reasoning process and the nature of its errors, we conducted detailed case studies of both scale-based and formula-based medical calculations. Appendix 4 and Appendix 5 present complete model responses for CURB-65 pneumonia severity scoring and serum osmolality calculation, respectively.

These case studies reveal that our RL-trained model demonstrates strong multi-step reasoning capabilities: it correctly identifies relevant clinical parameters, applies appropriate medical formulas or scoring criteria, and executes precise numerical computations. The model’s ability to maintain coherent reasoning chains while handling complex unit conversions and numerical precision requirements highlights the effectiveness of the MedCalc-Env training.

Based on our observations and the analysis presented in the supplementary material, we categorize the errors made by LLMs into four main types:

- **Type A: Knowledge Errors:** These occur when the model incorrectly recalls or applies a formula or rule. In zero-shot settings, this category typically accounts for the highest proportion of errors.
- **Type B: Extraction Errors:** These errors arise when the model fails to correctly identify or extract attribute values from the patient’s medical record. This often involves issues with natural language understanding, synonym recognition, or discerning relevant information from noisy contexts.
- **Type C: Calculation Errors:** These are arithmetic errors where the model correctly identifies the formula and extracts the values but makes mistakes during the numerical computation. This highlights a vulnerability in the execution phase of the calculation task.
- **Type D: Other Errors:** This category includes miscellaneous errors such as improper formatting, incorrect unit conversions, or non-standard date outputs.

Our comprehensive error analysis across the evaluation datasets indicates a significant shift in error distribution patterns. In zero-shot conditions, Type A (knowledge errors) constituted the majority of failures, indicating fundamental gaps in medical knowledge recall. However, after targeted interventions including medical knowledge prompting and especially after MedCalc-Env RL training, we observed a dramatic reduction in Type A errors, while Type B (extraction errors) and Type C (calculation errors) also decreased.

The MedCalc-Env RL training directly addresses these remaining challenges. By emphasizing accurate attribute extraction, medical knowledge recall, proper formula application, and error-free numerical computation, our approach specifically targets these bottlenecks, ultimately leading to more reliable and trustworthy medical calculation capabilities.

5 Related Work

5.1 LLMs in Medical Domain and Qualitative Benchmarks

The application of Large Language Models (LLMs) in medicine has seen rapid growth (Wang et al., 2025b; Xie et al., 2024; Labrak et al., 2024; Sallinen et al., 2025; Wu et al., 2025; Liu et al., 2024; Lai et al., 2025), primarily focusing on tasks that rely on natural language understanding and generation. Early and foundational benchmarks, such as **MedQA** (Jin & et al., 2023), **PubMedQA** (Jin & et al., 2019), and **MedMCQA** (Pal & et al., 2022), have been crucial in assessing LLMs's medical knowledge, diagnostic capabilities, and ability to follow clinical guidelines. These benchmarks typically involve multiple-choice questions, open-ended descriptive reasoning, and information retrieval, which are essential for evaluating the qualitative aspects of medical AI. More recent and holistic evaluation frameworks, like **HealthBench** (Arora et al., 2025) and the **Swedish Medical LLM Benchmark (SMLB)** (Moëll et al., 2025), aim to provide a more comprehensive assessment, including safety and ethical considerations. However, these efforts generally emphasize descriptive and diagnostic reasoning, often overlooking the critical need for precise quantitative skills in clinical practice.

5.2 Quantitative Reasoning and Medical Calculation Benchmarks

Quantitative reasoning, which involves precise numerical computation and logical application of formulas and rules, is indispensable for evidence-based medical decision-making. Clinical calculators, built upon established equations and rule-based scoring systems, are routinely used for risk assessment, drug dosage calculation, and physiological parameter estimation. **MedCalc-Bench** (Khandekar et al., 2023), introduced as the first systematic benchmark for clinical calculation scenarios, marked a significant step towards evaluating LLMs's capabilities in this area. It includes over 50 different medical calculators and provides a foundational framework for assessing numerical accuracy. However, as discussed in our introduction, MedCalc-Bench's coverage is limited, lacking comprehensive representation across various medical specialties and complex, multi-step calculation tasks. Recent works, such as **MedRaC** (Wang et al., 2025a), have focused on diagnosing and improving LLM performance in evidence-based medical calculations, highlighting the ongoing challenge of numerical sensitivity and reasoning chain robustness. Our **MedCalc-Eval** significantly expands this landscape by offering a substantially larger and more diverse set of calculation tasks, including a wider range of specialties and more complex rule-based systems, providing a more rigorous stress test for LLMs's quantitative reasoning.

5.3 Reinforcement Learning for Enhanced LLM Reasoning

The challenge of multi-step and complex reasoning in LLMs has led to the exploration of advanced training paradigms, particularly those involving interactive learning and feedback. The **InternBootcamp** (Li et al., 2025) framework, on which our training environment is based, represents a novel approach to continuous model evolution through interactive feedback. This closed-loop system allows models to acquire and refine complex reasoning abilities by integrating evaluation directly into the training process. In the medical domain, Reinforcement Learning (RL) is increasingly being leveraged to align LLMs with complex clinical protocols and improve reasoning. Works like **EHRMIND** (Lin et al., 2025) and **Baichuan-M2** (Team et al., 2025) demonstrate the application of RL to tasks like EHR-based reasoning and the emergence of reasoning capability through RL, respectively. Furthermore, the use of RL for structured medical reasoning, such as aligning with ACR Imaging Appropriateness Criteria, underscores the potential of this approach. Our **MedCalc-Env** builds upon this foundation, adapting the interactive and feedback-driven nature of InternBootcamp and RL to the specific challenges of medical calculation, enabling the model to learn robust formula selection, unit handling, and multi-step computational accuracy.

6 Conclusion

In this work, we have successfully integrated the medical clinical calculator question-answering task into general-purpose large language models through the InternBootcamp framework. Our experimental results unequivocally demonstrate that models trained using the InternBootcamp approach exhibit significantly enhanced capabilities in medical clinical calculator tasks. This further validates the feasibility and effectiveness of the "general-purpose and specialized integration" technical route, providing a reusable methodology and valuable reference for future practices in medical domain-specific integration.

We introduced **MedCalc-Eval**, a comprehensive and challenging benchmark that addresses the limitations of existing evaluation datasets by offering a significantly expanded scope of medical calculation tasks across numerous clinical specialties. Complementing this, we developed **MedCalc-Env**, a reinforcement learning environment designed to foster multi-step clinical reasoning and action planning in LLMs. The state-of-the-art performance achieved by our RL-trained Qwen2.5-32B model on both MedCalc-Eval and the existing MedCalc-Bench underscores the potential of our approach to bridge the quantitative reasoning gap in current LLMs.

Despite these advancements, our error analysis highlights several areas for future improvement. Challenges persist in tasks requiring precise unit conversion, handling complex multi-condition logic, and achieving nuanced context understanding from patient notes. These areas represent critical avenues for future research to further enhance the reliability and accuracy of AI-powered clinical decision support systems.

For future work, we envision several directions:

- **Dynamic, Realistic Patient Simulation:** A critical next step is to move beyond static, structured benchmark data towards dynamically generating realistic, free-text patient descriptions as model inputs. This involves creating simulated clinical narratives (e.g., physician progress notes, emergency department reports) where patient attributes, medical histories, and clinical findings are presented in natural, unstructured language, complete with redundancies, synonyms, and irrelevant information. Training and evaluating models on such data will significantly enhance their robustness in information extraction and their applicability to real-world clinical text.
- **Multi-modal Integration:** Exploring the integration of multi-modal inputs (e.g., medical

images, physiological signals) to enable more comprehensive and context-aware medical calculations.

- **Explainable AI (XAI):** Developing methods to make the LLM’s reasoning process for medical calculations more transparent and explainable, which is crucial for trust and adoption in clinical settings, especially when handling complex, narrative-style inputs.
- **Cross-lingual and Cross-cultural Generalization:** Extending the model’s capabilities to handle medical calculations across different languages, clinical guidelines, and healthcare systems, improving global applicability.

We believe that continued research in these areas will pave the way for more robust, accurate, and clinically applicable large language models, ultimately contributing to safer and more efficient healthcare delivery.

References

- Arora, R. K., Wei, J., Hicks, R. S., Bowman, P., Quiñonero-Candela, J., Tsimpourlas, F., Sharman, M., Shah, M., Vallone, A., Beutel, A., Heidecke, J., and Singhal, K. Healthbench: Evaluating large language models towards improved human health, 2025. URL <https://arxiv.org/abs/2505.08775>.
- Contributors, X. Xtuner: A toolkit for efficiently fine-tuning llm. <https://github.com/InternLM/xtuner>, 2023.
- DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., Gao, Z., Liu, A., Xue, B., Wang, B., Wu, B., Feng, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Ding, H., Xin, H., Gao, H., Qu, H., Li, H., Guo, J., Li, J., Wang, J., Chen, J., Yuan, J., Qiu, J., Li, J., Cai, J. L., Ni, J., Liang, J., Chen, J., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Zhao, L., Wang, L., Zhang, L., Xu, L., Xia, L., Zhang, M., Zhang, M., Tang, M., Li, M., Wang, M., Li, M., Tian, N., Huang, P., Zhang, P., Wang, Q., Chen, Q., Du, Q., Ge, R., Zhang, R., Pan, R., Wang, R., Chen, R. J., Jin, R. L., Chen, R., Lu, S., Zhou, S., Chen, S., Ye, S., Wang, S., Yu, S., Zhou, S., Pan, S., Li, S. S., Zhou, S., Wu, S., Ye, S., Yun, T., Pei, T., Sun, T., Wang, T., Zeng, W., Zhao, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Xiao, W. L., An, W., Liu, X., Wang, X., Chen, X., Nie, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yang, X., Li, X., Su, X., Lin, X., Li, X. Q., Jin, X., Shen, X., Chen, X., Sun, X., Wang, X., Song, X., Zhou, X., Wang, X., Shan, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhang, Y., Xu, Y., Li, Y., Zhao, Y., Sun, Y., Wang, Y., Yu, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Ou, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Xiong, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Zhu, Y. X., Xu, Y., Huang, Y., Li, Y., Zheng, Y., Zhu, Y., Ma, Y., Tang, Y., Zha, Y., Yan, Y., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Xie, Z., Zhang, Z., Hao, Z., Ma, Z., Yan, Z., Wu, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Pan, Z., Huang, Z., Xu, Z., Zhang, Z., and Zhang, Z. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL <https://arxiv.org/abs/2501.12948>.
- Jin, Q. and et al. Pubmedqa: A dataset for biomedical question answering. *arXiv preprint arXiv:1909.02221*, 2019.
- Jin, Q. and et al. Medqa: A dataset of medical question answering for medical large language models. *arXiv preprint arXiv:2305.09617*, 2023.

Khandekar, N., Jin, Q., Xiong, G., Dunn, S., Applebaum, S. S., Anwar, Z., Sarfo-Gyamfi, M., Safranek, C. W., Anwar, A. A., Zhang, A., Gilson, A., Singer, M. B., Dave, A., Taylor, A., Zhang, A., Chen, Q., and Lu, Z. Medcalc-bench: Evaluating large language models for medical calculations. *arXiv preprint arXiv:2306.02163*, 2023.

Labrak, Y., Bazoge, A., Morin, E., Gourraud, P.-A., Rouvier, M., and Dufour, R. Biomistral: A collection of open-source pretrained large language models for medical domains, 2024. URL <https://arxiv.org/abs/2402.10373>.

Lai, Y., Zhong, J., Li, M., Zhao, S., and Yang, X. Med-r1: Reinforcement learning for generalizable medical reasoning in vision-language models. *arXiv preprint arXiv:2503.13939*, 2025.

Li, P., Ye, J., Chen, Y., Ma, Y., Yu, Z., Chen, K., Cui, G., Li, H., Chen, J., Lyu, C., Zhang, W., Li, L., Guo, Q., Lin, D., Zhou, B., and Chen, K. Internbootcamp technical report: Boosting llm reasoning with verifiable task scaling, 2025. URL <https://arxiv.org/abs/2508.08636>.

Lin, J., Wu, Z., and Sun, J. Training llms for ehr-based reasoning tasks via reinforcement learning, 2025. URL <https://arxiv.org/abs/2505.24105>.

Liu, J., Wang, Y., Du, J., Zhou, J. T., and Liu, Z. Medcot: Medical chain of thought via hierarchical expert, 2024. URL <https://arxiv.org/abs/2412.13736>.

Moëll, B., Farenstam, F., and Beskow, J. Swedish medical llm benchmark (smlb): Development and evaluation of a framework for assessing large language models in the swedish medical domain. *Frontiers in Artificial Intelligence*, 8:1557920, 2025.

Pal, K. and et al. Medmcqa: A large-scale dataset for medical multiple-choice question answering. *arXiv preprint arXiv:2203.10090*, 2022.

Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. Qwen2.5 technical report, 2025. URL <https://arxiv.org/abs/2412.15115>.

Sallinen, A., Solergibert, A.-J., Zhang, M., Boyé, G., Dupont-Roc, M., Theimer-Lienhard, X., Boisson, E., Bernath, B., Hadhri, H., Tran, A., Rabbani, T., Brokowski, T., Group, M. M. D. W., Rudner, T. G. J., and Hartley, M.-A. Llama-3-meditron: An open-weight suite of medical llms based on llama-3.1. <https://openreview.net/forum?id=ZcD35zKuj0>, March 2025. GenAI4Health Poster; CC BY 4.0; Accessed: 2025-05-14.

Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., and Wu, C. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, pp. 1279–1297. ACM, March 2025. doi: 10.1145/3689031.3696075. URL <http://dx.doi.org/10.1145/3689031.3696075>.

Team, M., Dou, C., Liu, C., Yang, F., Li, F., Jia, J., Chen, M., Ju, Q., Wang, S., Dang, S., Li, T., Zeng, X., Zhou, Y., Zhu, C., Pan, D., Deng, F., Ai, G., Dong, G., Zhang, H., Tai, J., Hong, J., Lu, K., Sun, L., Guo, P., Ma, Q., Xin, R., Yang, S., Zhang, S., Mo, Y., Liang, Z., Zhang, Z., Cui, H., Zhu, Z., and Wang, X. Baichuan-m2: Scaling medical capability with large verifier system, 2025. URL <https://arxiv.org/abs/2509.02208>.

Wang, B., Xia, I., Zhang, Y., Wang, J., Ouyang, F., Han, S., Cohan, A., Yu, H., and Yao, Z. From scores to steps: Diagnosing and improving llm performance in evidence-based medical calculations, 2025a. URL <https://arxiv.org/abs/2509.16584>.

- Wang, B., Zhao, H., Zhou, H., Song, L., Xu, M., Cheng, W., Zeng, X., Zhang, Y., Huo, Y., Wang, Z., Zhao, Z., Pan, D., Kou, F., Li, F., Chen, F., Dong, G., Liu, H., Zhang, H., He, J., Yang, J., Wu, K., Wu, K., Su, L., Niu, L., Sun, L., Wang, M., Fan, P., Shen, Q., Xin, R., Dang, S., Zhou, S., Chen, W., Luo, W., Chen, X., Men, X., Lin, X., Dong, X., Zhang, Y., Duan, Y., Zhou, Y., Ma, Z., and Wu, Z. Baichuan-m1: Pushing the medical capability of large language models, 2025b. URL <https://arxiv.org/abs/2502.12671>.
- Wen, X., Liu, Z., Zheng, S., Ye, S., Wu, Z., Wang, Y., Xu, Z., Liang, X., Li, J., Miao, Z., Bian, J., and Yang, M. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms, 2025. URL <https://arxiv.org/abs/2506.14245>.
- Wu, J., Deng, W., Li, X., Liu, S., Mi, T., Peng, Y., Xu, Z., Liu, Y., Cho, H., Choi, C.-I., Cao, Y., Ren, H., Li, X., Li, X., and Zhou, Y. Medreason: Eliciting factual medical reasoning steps in llms via knowledge graphs, 2025. URL <https://arxiv.org/abs/2504.00993>.
- Xie, Q., Chen, Q., Chen, A., Peng, C., Hu, Y., Lin, F., Peng, X., Huang, J., Zhang, J., Keloth, V., Zhou, X., Qian, L., He, H., Shung, D., Ohno-Machado, L., Wu, Y., Xu, H., and Bian, J. Me llama: Foundation large language models for medical applications, 2024. URL <https://arxiv.org/abs/2402.12749>.
- Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., Liu, X., Lin, H., Lin, Z., Ma, B., Sheng, G., Tong, Y., Zhang, C., Zhang, M., Zhang, W., Zhu, H., Zhu, J., Chen, J., Chen, J., Wang, C., Yu, H., Song, Y., Wei, X., Zhou, H., Liu, J., Ma, W.-Y., Zhang, Y.-Q., Yan, L., Qiao, M., Wu, Y., and Wang, M. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL <https://arxiv.org/abs/2503.14476>.

Appendix

A Details of MedCalc-Eval	17
A.1 Dataset Details	17
A.2 Model Performance	17
A.3 Ablation Experiments	17
B Prompt templates of MedCalc-Eval & MedCalc-Env	19
B.1 Formula-based Question Prompt	19
B.2 Formula-based Question Prompt with Formula Explanation	19
B.3 Scale-based Question Prompt	20
B.4 Scale-based Question Prompt with Scoring Criteria	20
C Case Studies	21
C.1 Case of Scale-based Question	21
C.2 Case of Formula-based Question	22
D Implementation Details of MedCalc-Env	23
E Visual web tool for auditing MedCalc-Eval	24

A Details of MedCalc-Eval

In this section, we show the details of our MedCalc-Eval.

A.1 Dataset Details

Table 2: Summary Statistics of the MedCalc-Eval

	Formula-based	Scale-based	Indicators
Categories	132	27	-
Total Numbers	629	80	1432

A.1.1 Top 20 Categories in Formula-based or Scale-based Question

Table 3: Top 20 Categories in Formula-based or Scale-based Question

(a) Formula-based Question		(b) Scale-based Question		
Rank	Category	Count	Category	Count
1	Laboratory Medicine	36	Cardiovascular Diseases	18
2	Pulmonary Diseases	31	Obstetrics and Gynecology Diseases	10
3	Nephrology	31	Neurological Diseases	6
4	Pediatrics	30	Hematology and Oncology	5
5	Cardiology	28	Pulmonary Diseases	5
6	Respiratory Medicine	25	Anesthesia	5
7	Special Issues	24	Urogenital Diseases	3
8	Urogenital Diseases	24	Digestive System Diseases	3
9	Endocrine and Metabolic Disorders	24	Rehabilitation Medicine	2
10	Hematology	20	Hepatobiliary Diseases	2
11	Respiratory Medicine/ICU	16	Pediatric Science	2
12	Emergency Department	16	Otorhinolaryngology	2
13	Pharmacy	15	Psychiatric Disorders	2
14	Radiology	14	Respiratory Medicine	2
15	Internal Medicine	14	Emergency Medicine	1
16	Emergency Medicine	13	Special Issues	1
17	Pharmacology	12	Gastrointestinal Diseases	1
18	Hepatobiliary Diseases	11	Geriatrics	1
19	Anesthesiology	10	Emergency Department	1
20	Nutritional Diseases	9	Intensive Care Unit	1

A.1.2 Formula Input Indicators

A.2 Model Performance

A.3 Ablation Experiments

Table 4: Most Frequently Occurring Formula Input Indicators(Top 20)

Rank	Indicator Name	Frequency
1	Weight	88
2	Age	62
3	Height [cm]	28
4	Height	23
5	Heart Rate	17
6	Urine Creatinine [mg/dL]	17
7	Tidal Volume	15
8	Blood Creatinine [mg/dL]	12
9	Hemoglobin	11
10	Systolic Blood Pressure	10
11	PaCO ₂	9
12	Cardiac Output	9
13	Respiratory Rate	9
14	PaO ₂	9
15	Serum Creatinine	8
16	Blood Sodium	8
17	Specificity	7
18	Post-dialysis Blood Urea Nitrogen	7
19	Time	7
20	Prevalence	7

Table 5: Performance of LLMs on MedCalc-Eval compared to MedCalc-Bench

Model	MedCalc-Eval (%)	MedCalc-Bench (%)
DeepSeek-R1-32B	19.0	14.3
Qwen2.5-7B	13.1	9.7
Qwen3-8B	18.2	9.6
Qwen2.5-32B	25.4	21.5
Qwen3-235B-A22B	31.1	24.5
Qwen2.5-7B + MedCalc-Env RL	15.6	11.5
Qwen2.5-32B + MedCalc-Env RL	40.8	20.0

Table 6: Ablation study results of Qwen2.5-7B model

Model	MedCalc-Eval (%)	MedCalc-Bench (%)	AIME24 (%)
Qwen2.5-7B	13.1	9.7	16.7
Qwen2.5-7B + MedCalc-Env RL	15.6	11.5	20.0

B Prompt templates of MedCalc-Eval & MedCalc-Env

In this section, we provide the prompt templates² in the MedCalc-Eval & MedCalc-Env, including:

- the prompt template of the formula-based question with/without formula explanation;
- the prompt template for the scale-based question with/without scale scoring criteria.

B.1 Formula-based Question Prompt

The Prompt for Formula-based Question

Patient Information: {patient_info}

Please calculate {formula_name}, retain {indicator_precision} decimal places.

Let's think step by step and output the final answer within $xxx : xxx$. For example
" $BMI : 20.5$ ".

Formula-based Question Example

Patient Information: Red blood cell count $4730.5 \times 10^9 / L$, hemoglobin 14.5 g/dL .

Please calculate mean corpuscular hemoglobin (MCH), retain 2 decimal places.

Let's think step by step and output the final answer within $xxx : xxx$. For example
" $BMI : 20.5$ ".

B.2 Formula-based Question Prompt with Formula Explanation

The Prompt for Formula-based Question with Formula Explanation

{formula_name}

Calculation formula: {formula}

Formula Explanation: {formula_explanation}

Patient Information: {patient_info}

Please calculate {formula_name}, retain {indicator_precision} decimal places.

Let's think step by step and output the final answer within $xxx : xxx$. For example
" $BMI : 20.5$ ".

Formula-based Question Example

Glomerular Filtration Rate

Calculation formula: $175 \times \text{serum creatinine} (-1.154) \times \text{age} (-0.203) \times \text{sex}$ Formula Explanation:

Sex: Male: 1, Female: 0.742

Patient Information: Serum creatinine 4.42 mg/dL , age 67, female.

Please calculate glomerular filtration rate, retain 3 decimal places.

Let's think step by step and output the final answer within $xxx : xxx$. For example
" $BMI : 20.5$ ".

²The original prompts and examples are in Chinese, and we translated them into English for convenience.

B.3 Scale-based Question Prompt

The Prompt for Scale-based Question

Patient Information: {patient_info}

Please calculate {scale_name}.

Let's think step by step and output the final answer within `xxx : xxx`. For example
"BMI : 20.5".

Scale-based Question Example

Patient Information: Limbs/Pelvis: Femoral fracture; comminuted pelvic fracture; knee ligament rupture; Face: LeFort type III fracture (complete facial separation); Abdomen: Small intestine/bladder perforation; subcapsular rupture of liver/spleen; intraperitoneal hemorrhage $\leq 1000\text{ml}$; Chest: Extensive crush injury to the chest; complete aortic transection; Head and neck: Head injury-related headache/dizziness; cervical spine sprain without fracture; minor rupture of external jugular vein (blood loss $\leq 20\%$); thyroid contusion; Body surface: Second- or third-degree burns $\geq 90\%$ of body surface area.

Please calculate AIS score.

Let's think step by step and output the final answer within `xxx : xxx`. For example
"BMI : 20.5".

B.4 Scale-based Question Prompt with Scoring Criteria

The Prompt for Scale-based Question

{scale_name}

Scale scoring criteria: {scoring_criteria}

Patient Information: {patient_info}

Please calculate {scale_name}.

Let's think step by step and output the final answer within `xxx : xxx`. For example
"BMI : 20.5".

Scale-based Question Example

Spetsler-Martin grade for an intracranial arteriovenous malformation (AVM)

Scale scoring criteria: [AVMDiameter][SingleChoice]

$<3\text{ cm}$ (1 point) ; $3\text{-}6\text{ cm}$ (2 points) ; $>6\text{ cm}$ (3 points)

[AVMLocation][SingleChoice]

Non-functional area (0 point) ; Functional area (1 point)

[AVMDrainage][SingleChoice]

Superficial venous drainage (0 point) ; Deep venous drainage (1 point)

Patient Information: AVM Drainage: Superficial venous drainage, AVM Diameter: 3-6 cm, AVM Location: Non-functional area.

Please calculate Spetsler-Martin grade for an intracranial arteriovenous malformation (AVM).

Let's think step by step and output the final answer within `xxx : xxx`. For example
"BMI : 20.5".

C Case Studies

In this section, we provide the complete model responses³ for the case studies in Section 4.

C.1 Case of Scale-based Question

Case study: CURB-65 pneumonia severity score

Problem

Patient Information: Blood Urea Nitrogen: >19 mg/dL, Impaired Consciousness: Yes, Age: <65 years, Respiratory Rate: <30 breaths/min, Blood Pressure: Systolic <90 or Diastolic ≤ 60 mmHg. Please calculate the CURB-65 pneumonia severity score.

Let's think step by step and output the final answer within $\boxed{xxx : xxx}$. For example $\boxed{BMI : 20.5}$.

The CURB-65 pneumonia severity scoring system includes the following factors:

- C (Confusion, impaired consciousness): 1 point
- U (Uremia, blood urea nitrogen >20 mg/dL): 1 point
- R (Respiratory rate, respiratory rate ≥ 30 breaths/minute): 1 point
- B (Blood pressure, systolic blood pressure <90 mmHg or diastolic blood pressure ≤ 60 mmHg): 1 point
- Age ≥ 65 years: 1 point

Based on the information provided in the question, the patient's scoring criteria include:

- Impaired consciousness (present): 1 point
- Blood urea nitrogen: >19 mg/dL (the target blood urea nitrogen is >20 mg/dL), but the question does not specify >20 mg/dL, only >19 mg/dL, so a score of 1 is assumed. (Although it is below 20, the scoring scale is usually not subdivided, so a score of 1 is assumed for this question.)
- Respiratory rate: <30 breaths/minute (Not meeting the target of 30 beats/minute)
- Blood pressure: Systolic pressure <90 or diastolic pressure ≤ 60 mmHg (Not meeting the target)

The patient does not meet the "age ≥ 65 years" requirement.

Comprehensive calculation:

1. Impaired consciousness: 1 point
2. Blood urea nitrogen: 1 point
3. Respiratory rate: 0 point
4. Blood pressure: 1 point

Total score: $1 + 1 + 0 + 1 = 3$ points

Final answer: $\boxed{3}$

Figure 4: The complete model response for the scale-based question case study.

³The original model responses and questions are in Chinese, and we translated them into English for convenience.

C.2 Case of Formula-based Question

Case study: Osmol

Problem

Patient information: Glucose 297.0 mg/dL, Blood urea nitrogen 9.44 mg/dL, Sodium 1793.74 mEq/L. Please calculate the estimated osmolality (serum) – osmolality, retain 1 decimal places.

Let's think step by step and output the final answer within `xxx : xxx`. For example "`BMI : 20.5`".

To calculate serum osmolality (Osmol), use the formula:

$$Osmol = 2[Na^+] + [Glucose]/18 + [BUN]/2.8$$

, where the units must be consistent. Here, $[Na^+]$ is mEq/L, $[Glucose]$ is mg/dL, and $[BUN]$ is mg/dL. Based on the given data, let's perform the following calculations:

1. Glucose ($[Glucose] = 297.0$ mg/dL)
2. Blood urea nitrogen ($[BUN] = 9.44$ mg/dL)
3. Sodium ($[Na^+] = 1793.74$ mEq/L)

Substitute the values into the formula and calculate:

$$Osmol = 2[Na^+] + \frac{[Glucose]}{18} + \frac{[BUN]}{2.8}$$

$$Osmol = 2(1793.74) + \frac{297.0}{18} + \frac{9.44}{2.8}$$

$$Osmol = 3587.48 + 16.5 + 3.371428571$$

$$Osmol \approx 3607.351428571$$

Take the result to one decimal place:

$$Osmol \approx 3607.4$$

Thus, the estimated osmolality (serum) is `3607.4`.

Figure 5: The complete model response for the formula-based question case study.

D Implementation Details of MedCalc-Env

In this section, we show the implementation details of our **MedCalc-Env**, a novel reinforcement learning (RL) environment built upon the InternBootcamp framework⁴. InternBootcamp is an open-source framework comprising 1000+ domain-diverse task environments specifically designed for LLM reasoning research.

Implementation Details of MedCalc-Env

Create a Bootcamp class, inherit from the Basebootcamp class
 class MedCalculatorSandbox(Basebootcamp):

method to generate parameters (construct a single question or validate a response)

```
def _gen_a_case(self, category, name):
    details = self.config[category][name]
    ...
    return {
        "category": category,
        "name": name,
        "inputs": inputs,
        "add_rule": random.random() < self.add_rule_ratio,
        "target": target,
    }

def case_generator(self):
    category = random.choice(['equation', 'scale'])
    name = random.choice(list(self.config[category].keys()))
    return self._gen_a_case(category, name)
```

method to construct the problem statement for a single question

```
def prompt_func(self, case):
    indicators = self.config["indicator"]
    random.shuffle(case["inputs"])
    inp_items = ''.join(case["inputs"])
    out_item = case["name"]
    ...
    rule = self.rules[case["name"]] if case["add_rule"] else ""
    ques = f"{rule}Patient Information:{inp_items}. Please calculate{out_item}{other_item}."
    return ques + '\nLet's think step by step and output the final answer within \boxed{xxx:xxx}. For example \boxed{BMI:20.5}.'
```

method to validate whether the answer is correct

```
@classmethod
def _verify_correction(cls, solution, identity):
    if ':' in solution:
        solution = solution.split(':')[1].strip()
    elif '' in solution:
        solution = solution.split('')[-1].strip()

    return solution.strip() == str(identity['target'])
```

⁴InternBootcamp: <https://github.com/InternLM/InternBootcamp>

E Visual web tool for auditing MedCalc-Eval

In this section, we demonstrate a web tool we developed that is convenient for auditing our MedCalc-Eval dataset. It allows for easy previewing of each question in the dataset, error alerts, and statistical results. The interface and functionality are shown below:

Figure 6: **Preview page.** This is where you can view detailed data for each class of question.

Figure 7: **Statistical results.** This is used to view the overall statistics of the dataset, including the type and number of questions, and the number of errors.

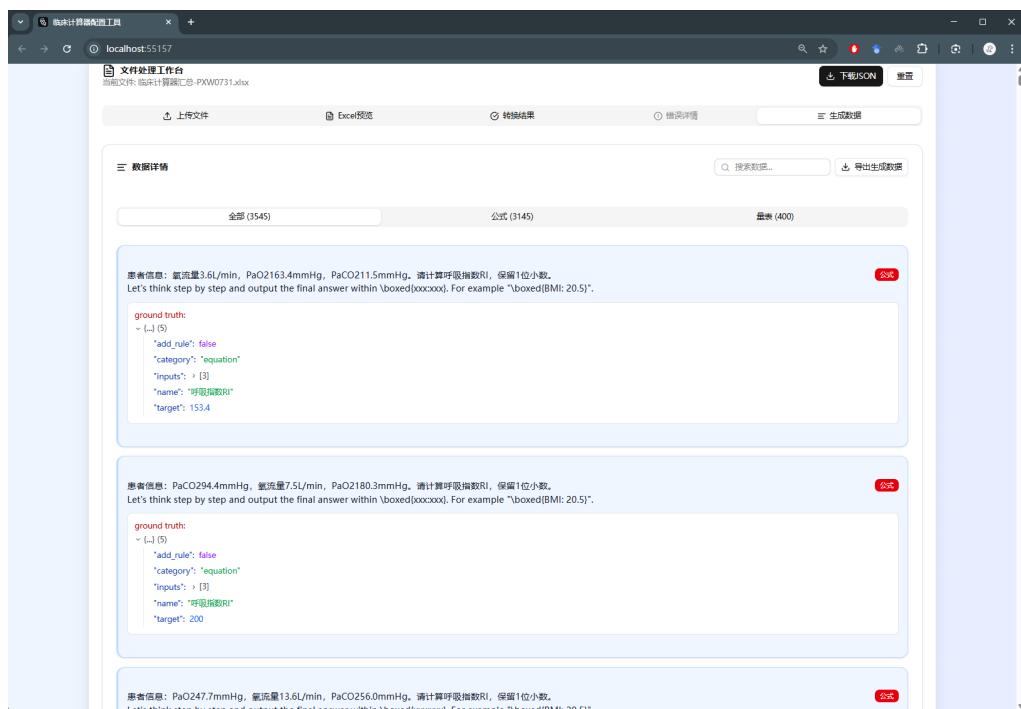


Figure 8: **Generated data.** This is used to preview and review the generated data, and indicate any potential errors.