

Beyond Correctness: Validity-Oriented Evaluation of Biomedical LLM Judges

Rodrigo de Oliveira
Independent Researcher*, London, UK
rdeoliveira.phd@gmail.com

Federico Pittino
IQVIA, Barcelona, Spain
federico.pittino@iqvia.com

James Gwinnutt
IQVIA, Reading, UK
james.gwinnutt@iqvia.com

Jay Nanavati
IQVIA, Cambridge, UK
jay.nanavati@iqvia.com

Abstract

We propose a scalable, validity-oriented pipeline for evaluating biomedical LLM judges when high-quality human judgments are scarce. First, we augment existing human-labelled biomedical benchmarks with deterministic, metric-grounded mutations that produce auditable preference pairs. Second, we evaluate judges beyond aggregate correctness using three deployment-relevant dimensions: correctness against metric-derived gold labels, robustness under repeated stochastic sampling, and compliance with the requested output format. We use this pipeline to assess Llama-3.1-8B-Instruct under four regimes: (1) *base*, using the instruct model as is; (2) *SFT*, distillation-based supervised fine-tuning only; (3) *RL*, GRPO-based reinforcement learning only; and (4) *SFT*→*RL*, SFT followed by RL. The base and single-stage regimes struggle on structured medical discrimination such as PICO extraction and clinical calculations, whereas SFT→RL performs best across correctness, compliance, and robustness; gains concentrate on decomposable tasks (PICO, MedCalc), at times matching or outperforming frontier models.

1 Introduction

LLM-as-a-judge is increasingly used to scale evaluation beyond human ratings [1]. In high-stakes settings such as healthcare, however, judges must remain valid in real-world use. Biomedical applications—from structured extraction and clinical summarization [2] to clinical interviews [3] and exam-style question answering [4, 5]—require fine-grained discrimination of PICO spans, calculation steps, and subtle distractors.

Most existing work treats agreement with human or expert judgments as the primary quality indicator [1, 6, 7], giving less attention to operational failures. A disagreement may reflect a substantive error, but may also arise from verdict instability or a non-parseable response [8, 9]. We therefore introduce a three-axis protocol measuring criterion-linked correctness, robustness, and compliance, and ask:

RQ1: *How often are judge disagreements operational in biomedical tasks?*

We compare small and frontier judges across biomedical tasks. Frontier models provide a strong ceiling, but fine-tuning small models may offer a more cost-effective route to specialized evaluators.

*This work was largely carried out while the author was employed at IQVIA.

Recent work trains thinking judges with supervised fine-tuning or reinforcement learning [10–15], typically via GRPO [16]; we therefore ask:

RQ2: *What training regime, if any, works best for small LLM judges in biomedical tasks?*

Training and evaluating these judges requires preference data—i.e. pairs of chosen/rejected responses—that would be costly to obtain through new human-rating campaigns. We therefore follow the approach of deterministic synthetic-data generation with verifiable ground truth [2–5, 17]. In real-world pipelines, strong generators often produce very good responses, thereby increasing the difficulty for LLM judges: it is arguably easier to tell a good from a bad response, than a good from a great response. As such, it is highly desirable to adjust the difficulty of synthetic pairs at the point of generation, motivating our final question:

RQ3: *Can deterministic data generation model difficulty in partially correct biomedical tasks?*

Thus, our contributions are three-fold:

1. A three-axis, validity-oriented **evaluation protocol** for biomedical LLM judges, combining criterion-linked correctness with robustness under resampling and compliance.
2. A deterministic and scalable method for augmenting biomedical benchmarks with auditable, metric-grounded **synthetic preference data** for judge training and evaluation.
3. Finally, an **empirical study of** four distinct **training** regimes of **small LLMs** using the proposed protocol and augmented data.

2 Method

2.1 Biomedical tasks

We train and evaluate on five biomedical tasks² grouped into two families by response structure. *Decomposable* tasks—PICO span extraction and MedCalc—require judging partially correct outputs: whether extracted trial elements or clinical calculations are right. *Atomic* tasks—MedBullets, MedQA, and PubMedQA—require judging all-or-nothing exam-style answers where one option is fully correct.

2.2 Synthetic data generation

Every preference label is traceable to ground truth and a task metric. Algorithms 1 and 2 in Appendix A implement candidate generation, pair selection, and the suite-specific operator o and metric m for each benchmark. For each source example with clinical text x and ground truth y , a task-specific mutation operator o generates K candidate responses:

$$\{c_1, \dots, c_K\} = o(x, y). \quad (1)$$

On PICO, o adds or deletes spans; on MedCalc, it perturbs numeric inputs; on atomic tasks, it substitutes multiple-choice options (Appendix A).

Each candidate is scored against ground truth and ranked by quality (Algorithm 1):

$$s_k = m(c_k, y), \quad (2)$$

Here, m is macro-F1 on PICO, MSE on MedCalc, and binary correctness on atomic tasks.

From the ranked pool, preprocessing selects one valid chosen–rejected pair per source example (Algorithm 2). Each retained *pair* is presented in slots A and B with gold verdict $l \in \{A, B\}$ naming the preferred slot:

$$l = \begin{cases} A & \text{if } s_A > s_B, \\ B & \text{otherwise,} \end{cases} \quad (3)$$

²PICO — https://huggingface.co/datasets/bigbio/ebm_pico; MedCalc — <https://huggingface.co/datasets/ncbi/MedCalc-Bench-v1.2>; MedBullets — <https://huggingface.co/datasets/super-dainiu/medagents-benchmark>; MedQA — <https://huggingface.co/datasets/super-dainiu/medagents-benchmark>; PubMedQA — <https://huggingface.co/datasets/super-dainiu/medagents-benchmark>.

where s_A and s_B are the candidate scores. On decomposable tasks, *pairs* are drawn from the ranking with integer rank gap

$$\delta = j - i, \quad (4)$$

where $i < j$ are the rank positions (smaller δ = harder *pair*); we stratify decomposable pairs into three per-example δ tertiles and drop pairs whose scores are too close. On atomic tasks, *pairs* contrast correct and incorrect options without rank-gap stratification.

SFT data. A frontier teacher (Claude Opus 4.6³) samples judge generation g , which receives a deterministic quality score Q :

$$Q(g, l) = f_{\text{fmt}}(g) + f_{\text{parse}}(g) + f_{\text{align}}(g, l) \in \{0, 1, 2, 3\}, \quad (5)$$

where f_{fmt} and f_{parse} require valid J1 judge format and a parseable verdict or score. The SFT corpus distills both pairwise verdict and pointwise score prompts from the same pairs. For pairwise examples, $f_{\text{align}}=1$ iff the teacher’s A/B verdict matches the metric-derived gold verdict l . For pointwise examples, the teacher scores each candidate in a preference pair independently; $f_{\text{align}}=1$ iff its score on the chosen candidate exceeds its score on the rejected candidate.

RL data. RL uses the same pairs and gold verdicts to provide verifiable J1-style rewards [10]; no teacher completions are stored. Because judges are sensitive to candidate order [1], we randomly swap slots during training and evaluation.

The pipeline scales with existing benchmarks rather than new rating campaigns, and its fixed operators and metrics make every pair auditable and replayable.

2.3 Training

All experiments use Llama-3.1-8B-Instruct on synthetic pairs from the five-suite pipeline: $\sim 27\text{K}$ training preference pairs for RL and SFT $\rightarrow$ RL, and $\sim 48\text{K}$ distilled teacher completions with $Q=3$ (pairwise and pointwise) for SFT; $\sim 3\text{K}$ pairs are held out for meta-evaluation (§ 2.4). Judge prompt templates are in Appendix B. We implement SFT and RL with Hugging Face’s TRL package⁴. We compare four regimes:

- **Base.** The Llama-3.1-8B-Instruct checkpoint is used as is, without additional training.
- **SFT.** Supervised fine-tuning on distilled teacher prompt–completion pairs that pass the maximum quality filter ($Q=3$), mixing pairwise and pointwise judge modes. The model is trained with completion-only cross-entropy on teacher completions; prompt tokens are masked from the loss.
- **RL.** Pairwise judge reinforcement learning from the instruct checkpoint, without teacher completions, following J1 [10] with verifiable pairwise rewards optimized with GRPO [16]. On each preference pair the student samples a judge completion; the reward is 1 when the parsed A/B verdict matches the gold verdict l , and 0 otherwise.
- **SFT $\rightarrow$ RL.** The SFT checkpoint is further tuned with the same pairwise RL stage on the same preference pairs and reward.

All tuning stages use LoRA adapters ($r=16$, $\alpha=32$) on attention and MLP projections. SFT runs for 3 epochs with learning rate 2×10^{-5} , per-device batch size 2, and 8-step gradient accumulation on one NVIDIA A100 GPU. RL runs for 1 epoch with learning rate 1×10^{-6} , five rollouts per prompt, and KL coefficient $\beta=0.01$ (J1 defaults for 8B models) on eight NVIDIA H100 GPUs.

2.4 Meta-evaluation

We meta-evaluate every checkpoint in pairwise mode on a held-out split ($\sim 3\text{K}$ comparisons across five tasks), judging each comparison three times with stochastic decoding (`num_repetitions=3`). Correctness is agreement with the metric-derived gold verdict l from Eqs. (3)–(4); we additionally report robustness under resampling and compliance when the judge errs. Related work studies bias,

³<https://www-cdn.anthropic.com/14e4fb01875d2a69f646fa5e574dea2b1c0ff7b5.pdf>

⁴<https://huggingface.co/docs/trl>

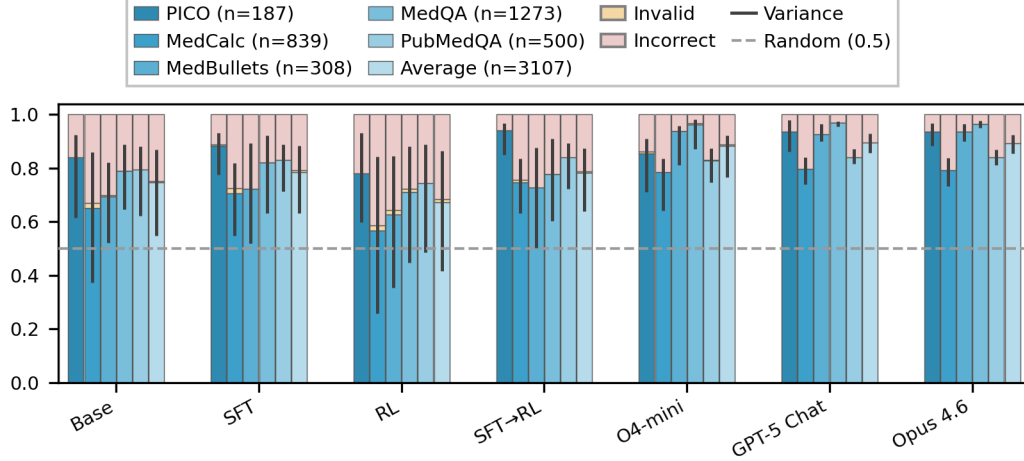


Figure 1: Response rates (correct, invalid, and incorrect) **by dataset**. Each dataset compares base, tuned, and frontier judge models (n in legend). **Variance** whiskers show robustness of correct response rate (Eq. (6)). Exact values in Appendix C.

consistency, or multi-criterion assessment separately [2, 8, 9, 18]; we combine correctness with two deployment-oriented diagnostics under one verifiable protocol.

Untuned frontier APIs—O4-mini⁵, GPT-5 Chat⁶, and Opus 4.6³—serve as strong baselines.

Correctness. Pairwise correctness is majority vote over the three parsed A/B verdicts. We slice it by response type—decomposable (PICO, MedCalc) versus atomic QA—and, for decomposable tasks, by three rank-gap tertiles. This is the axis used in most prior judge benchmarks [1, 6, 7].

Robustness. A robust judge should produce the same rating under repeated stochastic decoding. For each comparison we derive majority-vote, all-sample, and any-sample correctness. Let a_{maj} , a_{all} , and a_{any} denote their dataset means. We report verdict variance as the mean whisker span

$$v = |a_{\text{any}} - a_{\text{all}}|, \quad (6)$$

averaged across datasets (equivalently, the gap between lenient any-vote and strict all-vote correctness); lower v indicates less flip-flopping under resampling. This measures resampling robustness without human re-annotation, analogous in aim to Sage [9] but defined on repeated verdicts over fixed gold preferences.

Compliance. We separate substantive discrimination errors from format failures. For each comparison we partition outcomes into parseable wrong verdicts (*legitimate errors*) and completions without a parseable A/B verdict (*invalid responses*). This matters operationally because a non-compliant judge cannot be consumed by an automated pipeline. Compliance failure is the invalid share among all errors,

$$\rho = \frac{n_{\text{invalid}}}{n_{\text{invalid}} + n_{\text{legitimate}}}, \quad (7)$$

with lower ρ indicating that mistakes are more often wrong-but-parseable judgments than brittle non-responses. Unlike bias-stress benchmarks that perturb inputs [8], this axis measures compliance conditional on error—whether a wrong judge still returns an actionable verdict.

Together, the axes assess criterion-linked correctness, reproducibility, and actionable output.

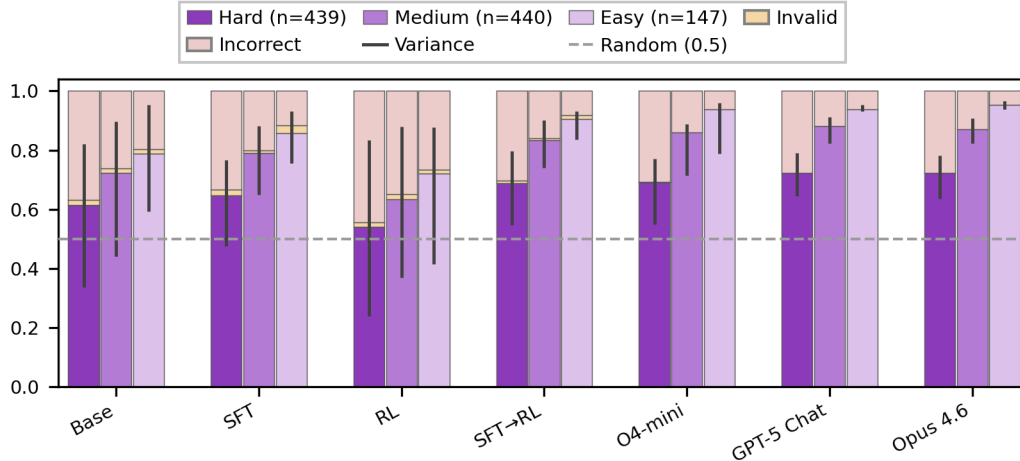


Figure 2: Response rates (correct, invalid, and incorrect) **by rank-gap difficulty**. Decomposable items only (PICO + MedCalc), stratified by **Hard**, **Medium**, and **Easy** tertiles of δ per source example (hard = smallest gaps; n in legend). **Variance** whiskers show robustness of correct response rate (Eq. (6)). Exact values in Appendix C.

3 Results

RQ1: *How often are judge disagreements operational in biomedical tasks?*

Figure 1 shows that not every disagreement is a legitimate judging error: some failures are operational, arising from verdict instability under resampling or from non-parseable responses. Among trained regimes, SFT→RL reduces the average invalid-response share to 1.6% versus 3.6% for RL, while its verdict variance is 0.05 versus 0.08 for base; the exact values are reported in Tables 4 and 3. Thus, operational failures are measurable and non-negligible, but can be reduced through training.

RQ2: *What training regime, if any, works best for small LLM judges in biomedical tasks?*

Applying the validity-oriented evaluation protocol (§ 2.4), SFT→RL beats all other trained regimes on correctness, compliance, and robustness. SFT→RL achieves the highest average correctness (0.81 pairwise correctness; Figure 1). Gains concentrate on decomposable items (+0.09 on PICO and MedCalc) rather than atomic QA (+0.02). Applying RL directly to the base model without SFT is harmful (−0.06 versus base), which suggests biomedical judge RL needs a domain prior. On individual suites, SFT→RL matches or beats frontier judges on PICO and PubMedQA, but frontier models still lead on average and clearly dominate atomic multiple-choice discrimination. Thus, for small biomedical judges, training order matters: RL benefits from the domain prior established by SFT.

RQ3: *Can deterministic data generation model difficulty in partially correct biomedical tasks?*

On decomposable tasks, rank-gap tertiles successfully model difficulty: pairwise correctness increases monotonically from hard to easy bins (Figure 2; e.g., teach-then-incentivising: 0.69 → 0.83 → 0.90). SFT→RL gains hold at every tertile, including close-call pairs in the hardest tier. Thus, deterministic preference construction can make synthetic evaluation more realistic by exposing judges to calibrated close-call comparisons.

⁵<https://developers.openai.com/api/docs/models/o4-mini>

⁶<https://developers.openai.com/api/docs/models/gpt-5-chat-latest>

4 Limitations

First, our experiments use one base model, Llama-3.1-8B-Instruct. We also attempted Qwen models, but many generations were non-compliant when using Hugging Face’s TRL package; reporting correctness or compliance figures from those runs would therefore be misleading, so we omit them.

Second, hardware constraints limited our study to an 8B model. Larger models, such as 70B checkpoints, may behave differently and are an important direction for future work.

Third, our synthetic data pipeline is correctness-centric: it deterministically injects errors into otherwise grounded responses. The same machinery could be extended to safety-centric stress tests by deterministically injecting dangerous content into rejected responses, either instead of or alongside ordinary correctness errors.

Finally, we are aware that post-hoc calibration can improve trustworthiness in biomedical language-model applications [17], but we have not yet tested calibration methods for LLM judges.

5 Related work

Recent work trains generative judges with RL on synthetic preferences and verifiable rewards [10–14], typically via GRPO [16]. We do not propose a new optimizer; we transfer J1 [10] to biomedicine and test whether domain SFT should precede RL. General judge benchmarks [1, 6, 7] and tuned evaluators such as GLIDER [15] standardize open-domain chat and reasoning; our study is domain-specific and explicitly contrasts decomposable structured tasks, where partial correctness matters [17], with atomic QA. Biomedical and clinical judging [2, 3, 18, 19] remains largely prompting- or SFT-based rather than pairwise judge RL [10].

Prior work pursues complementary judge-assessment axes: EvalBiasBench stress-tests robustness to superficial biases [8]; Sage measures local and global preference consistency without human gold [9]; and YESciEval and clinical-judge studies employ multi-rubric or validated instruments [2, 18]. These lines of work typically target different settings and remain separate from tuning studies. We contribute a three-dimensional, validity-oriented evaluation protocol (correctness, robustness, compliance) on fully verifiable, metric-derived biomedical preferences. To our knowledge, no prior work combines domain SFT, RL judge tuning, and systematic pairwise meta-evaluation along these axes across structured biomedical tasks with metric-derived reference labels.

6 Conclusion

In this study, we introduce a validity-oriented pipeline combining metric-grounded synthetic preferences with correctness, robustness, and compliance. We use it to compare four regimes across five biomedical benchmarks and to examine difficulty in partially correct tasks.

We answer RQ1 by showing that operational failures are distinct from legitimate judging errors: among trained regimes, SFT→RL reduces the average invalid-response share to 1.6% versus 3.6% for RL, while its verdict variance is 0.05 versus 0.08 for base. These measures show why judge evaluation should extend beyond correctness alone.

We answer RQ2 by finding that SFT→RL is the strongest tested regime across correctness, robustness, and compliance. Its gains concentrate on decomposable tasks, reaching 0.84 versus 0.75 for base, while direct RL from the base model is harmful; frontier models nevertheless remain stronger on average, especially on atomic question answering.

We answer RQ3 affirmatively: on decomposable tasks, pairwise correctness rises monotonically across rank-gap tertiles, from 0.69 in the hardest bin to 0.90 in the easiest for SFT→RL, with its gains holding at every difficulty level. This supports using deterministic, metric-grounded pairs to study the close-call comparisons that arise when strong generators produce similarly good candidates.

Together, these results show how auditable preference data and multidimensional evaluation can make small biomedical LLM judges more reliable and operationally useful.

References

- [1] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Neural Information Processing Systems*, 2023. URL <https://mlanthology.org/neurips/2023/zheng2023neurips-judging/>.
- [2] Emma Croxford, Yanjun Gao, Elliot First, Nicholas Pellegrino, Miranda Schnier, John Caskey, Madeline Oguss, Graham Wills, Guanhua Chen, Dmitriy Dligach, Matthew M. Churpek, Anoop Mayampurath, Frank Liao, Cherodeep Goswami, Karen K. Wong, Brian W. Patterson, and Majid Afshar. Evaluating clinical AI summaries with large language models as judges. *npj Digital Medicine*, 8(1):640, November 2025. doi: 10.1038/s41746-025-02005-2. URL <https://www.nature.com/articles/s41746-025-02005-2>.
- [3] Jing Chen, Zhihua Wei, Wei Zhang, Yingying Hu, and Qiong Zhang. CliniChat: A multi-source knowledge-driven framework for clinical interview dialogue reconstruction and evaluation. *arXiv preprint arXiv:2504.10418*, 2025. URL <https://arxiv.org/abs/2504.10418>.
- [4] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, July 2021. doi: 10.3390/app11146421. URL <https://www.mdpi.com/2076-3417/11/14/6421>.
- [5] Anastasios Nentidis, Georgios Katsimpras, Anastasia Krithara, Martin Krallinger, Miguel Rodríguez-Ortega, Eduard Rodríguez-López, Natalia Loukachevitch, Andrey Sakhovskiy, Elena Tutubalina, Dimitris Dimitriadis, Grigorios Tsoumakas, George Giannakoulas, Alexandra Bekiaridou, Athanasios Samaras, Giorgio Maria Di Nunzio, Nicola Ferro, Stefano Marchesin, Marco Martinelli, Gianmaria Silvello, and Georgios Paliouras. Overview of BioASQ 2025: The thirteenth BioASQ challenge on large-scale biomedical semantic indexing and question answering. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 173–198. Springer Nature Switzerland, September 2025. ISBN 978-3-032-04354-2. doi: 10.1007/978-3-032-04354-2_12. URL https://link.springer.com/chapter/10.1007/978-3-032-04354-2_12.
- [6] Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. RewardBench: Evaluating reward models for language modeling. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1755–1797, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.96. URL <https://aclanthology.org/2025.findings-naacl.96/>.
- [7] Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Yuan Tang, Alejandro Cuadron, Chengguang Wang, Raluca Ada Popa, and Ion Stoica. JudgeBench: A benchmark for evaluating LLM-based judges. In *International Conference on Learning Representations*, 2025. URL <https://mlanthology.org/iclr/2025/tan2025iclr-judgebench/>.
- [8] Junsoo Park, Seungyeon Jwa, Meiying Ren, Daeyoung Kim, and Sanghyuk Choi. OffsetBias: Leveraging debiased data for tuning evaluators. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1043–1067, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.57. URL <https://aclanthology.org/2024.findings-emnlp.57/>.
- [9] Yuanning Feng, Sinan Wang, Zhengxiang Cheng, Yao Wan, and Dongping Chen. Are we on the right way to assessing LLM-as-a-judge? *arXiv preprint arXiv:2512.16041*, 2025. URL <https://arxiv.org/abs/2512.16041>.
- [10] Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. J1: Incentivizing thinking in LLM-as-a-judge via reinforcement learning. In *International Conference on Learning Representations*, 2026. URL <https://mlanthology.org/iclr/2026/whitehouse2026iclr-j1/>.

- [11] Austin Xu, Yilun Zhou, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. J4R: Learning to judge with equivalent initial state group relative policy optimization. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1492–1511, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.67. URL <https://aclanthology.org/2026.acl-long.67/>.
- [12] Hui Huang, Yancheng He, Hongli Zhou, Rui Zhang, Wei Liu, Weixun Wang, Jiaheng Liu, and Wenbo Su. Think-J: Learning to think for generative LLM-as-a-judge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 31158–31166, March 2026. doi: 10.1609/aaai.v40i37.40377. URL <https://ojs.aaai.org/index.php/AAAI/article/view/40377>.
- [13] Nuo Chen, Zhiyuan Hu, Qingyun Zou, Jiaying Wu, Qian Wang, Bryan Hooi, and Bingsheng He. JudgeLRM: Large reasoning models as a judge. *arXiv preprint arXiv:2504.00050*, 2025. URL <https://arxiv.org/abs/2504.00050>.
- [14] Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, Hanghang Tong, and Heng Ji. RM-R1: Reward modeling as reasoning. In *International Conference on Learning Representations*, 2026. URL <https://mlanthology.org/iclr/2026/chen2026iclr-rmr1/>.
- [15] Darshan Deshpande, Selvan Sunitha Ravi, Sky CH Wang, Bartosz Mielczarek, Anand Kannappan, and Rebecca Qian. GLIDER: Grading LLM interactions and decisions using explainable ranking. *arXiv preprint arXiv:2412.14140*, 2024. URL <https://arxiv.org/abs/2412.14140>.
- [16] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL <https://arxiv.org/abs/2402.03300>.
- [17] Rodrigo de Oliveira, Matthew Garber, James M. Gwinnutt, Emaan Rashidi, Jwu-Hsuan (Shantina) Hwang, William Gilmour, Jay Nanavati, Khaldoun Zine El Abidine, and Christina DeFilippo Mack. A study of calibration as a measurement of trustworthiness of large language models in biomedical natural language processing. *JAMIA Open*, 8(4):ooaf058, 2025. doi: 10.1093/jamiaopen/ooaf058. URL <https://academic.oup.com/jamiaopen/article/8/4/ooaf058/8196848>.
- [18] Jennifer D’Souza, Hamed Babaei Giglou, and Quentin Münch. YESciEval: Robust LLM-as-a-judge for scientific question answering. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13749–13783, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.675. URL <https://aclanthology.org/2025.acl-long.675/>.
- [19] Ziyu Wang, Hao Li, Di Huang, Hye-Sung Kim, Chae-Won Shin, and Amir M. Rahmani. HealthQ: Unveiling questioning capabilities of LLM chains in healthcare conversations. *Smart Health*, 36:100570, June 2025. doi: 10.1016/j.smhl.2025.100570. URL <https://doi.org/10.1016/j.smhl.2025.100570>.
- [20] Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947. doi: 10.1007/BF02295996.

Appendix

A Synthetic data generation

This section details the synthetic pairwise pipeline from §2.2. For each source example with clinical text x and ground truth y , a task-specific mutation operator o proposes K candidate responses, each candidate is scored against y with a suite-specific metric m , and the pool is ranked by decreasing quality (ascending mean squared error on MedCalc). More formally:

Algorithm 1

```

GENERATE-CANDIDATES( $x, y, o, m, K$ )
1  let  $C$  and  $S$  be empty arrays //  $C$ : candidates;  $S$ : scores
2  for  $k = 1$  to  $K$  // Eq. (1)
3     $c = o(x, y)$ 
4     $s = m(c, y)$  // Eq. (2)
5    append  $c$  to  $C$ ; append  $s$  to  $S$ 
6  if  $m == \text{MSE}$ 
7    sort  $C$  and  $S$  so  $S[1] \leq S[2] \leq \dots \leq S[K]$ 
8  else sort  $C$  and  $S$  so  $S[1] \geq S[2] \geq \dots \geq S[K]$ 
9  return  $C, S$ 

```

The same pipeline applies to all benchmark suites; the distinction among them reduces to the shape of y , which dictates o and m (Table 1). Decomposable tasks (PICO, MedCalc) carry structured or numeric ground truth and later use rank-gap stratification when forming pairs; atomic multiple-choice tasks use boolean correctness and contrast one correct against one incorrect candidate. MedQA and PubMedQA follow the same atomic operator and metric as MedBullets; one QA example below suffices to illustrate that family.

Suite	Type	y	o	m	s_k
PICO	Decomposable	Span sets (P/I/C/O)	Add/delete spans	macro-F1 exact	$[0, 1] \uparrow$
MedCalc	Decomposable	Scalar (clearance)	Perturb formula inputs	MSE	$\geq 0 \downarrow$
QA tasks	Atomic	Multiple-choice label	Swap wrong option	boolean	$\{0, 1\}$

Table 1: Task-specific operators, metrics, and ground-truth types (Algorithms 1 and 2). MedQA and PubMedQA use the same atomic row as MedBullets.

Once candidates are ranked, Algorithm 2 bins all valid chosen–rejected pairs for a source example. δ -bin boundaries depend on K and the number of bins N (even splits of rank gaps in $\{1, \dots, K-1\}$; we use $N=3$ tertiles); scores enter only via the gap threshold $\tau=0.05$. On atomic tasks ($m=\text{boolean}$), every pair contrasts $C[1]$ with one lower rank and is placed in a single bin; on decomposable tasks, each rank gap δ fixes its bin b before any candidate index is visited, and scores only gate whether a pair is kept ($|S[i] - S[j]| > \tau$). Each stored tuple is $(chosen, rejected)$, where each side is a triple $(i, C[i], S[i])$ recording rank index, candidate, and score. Downstream preprocessing then draws one training example per source from these bins. More formally:

Algorithm 2

```
BIN-PREFERENCE-PAIRS( $C, S, m, K, N$ )
  //  $N$ : number of  $\delta$ -difficulty bins
  1 let  $B$  be an array of empty arrays // bins of ( $chosen, rejected$ ) triples
  2  $\tau = 0.05$  // score-gap threshold
  3 if  $m == \text{boolean}$ 
  4   let  $B[1]$  be an empty array
  5   for  $j = 2$  to  $K$ 
  6      $chosen = (1, C[1], S[1]); rejected = (j, C[j], S[j])$  // Eq. (3)
  7     append ( $chosen, rejected$ ) to  $B[1]$ 
  8 else compute  $\delta$ -bin boundaries from  $K$  and  $N$ 
  9   for  $b = 1$  to  $N$ 
  10    let  $B[b]$  be an empty array
  11    for  $\delta = 1$  to  $K - 1$  // Eq. (4)
  12       $b = \delta$ -bin of  $\delta$  given boundaries
  13      for  $i = 1$  to  $K - \delta$ 
  14         $j = i + \delta$ 
  15        if  $|S[i] - S[j]| > \tau$ 
  16           $chosen = (i, C[i], S[i]); rejected = (j, C[j], S[j])$  // Eq. (3)
  17          append ( $chosen, rejected$ ) to  $B[b]$ 
  18 return  $B$ 
```

Algorithm 2 organizes all valid preference pairs into difficulty-stratified bins; exact training examples are then selected by drawing one tuple per source and balancing underrepresented bins across the corpus (fixed random seed for reproducibility). When training prompts are assembled, the chosen and rejected candidates are randomly assigned to slots A and B (50% swap rate), again under a fixed seed, so gold verdicts from Eq. (3) remain reproducible.

Below we illustrate the full pipeline with one worked example per family: PICO, MedCalc, and MedBullets as the representative atomic QA task.

A.1 PICO

Split=test, ID=10070173.

Source text x :

Comparison of budesonide Turbuhaler with budesonide aqua in the treatment of seasonal allergic rhinitis. Rhinocort Study Group. OBJECTIVE To compare the effect of budesonide Turbuhaler 400 microg/day with budesonide aqua 256 microg/day in the treatment of seasonal allergic rhinitis (SAR). Secondly to ascertain patients' preferences for the two nasal devices and to assess quality of life. DESIGN Randomized, multicentre, double-blind, double-dummy, parallel groups study. SETTING Private practices and hospital clinics in Ontario, Quebec and Manitoba. POPULATION Two hundred and eighty-four out-patients with SAR, who were symptomatic during the ragweed season, volunteered for enrolment (243 randomized). RESULTS Mean daily nasal symptom scores were significantly reduced with treatment. There were no statistically significant changes from baseline for eye symptoms. Most patients (more than 80%) achieved substantial control of their symptoms with budesonide. The most common nasal and non-nasal adverse events for both groups were epistaxis and headache. Turbuhaler was easier to use and more convenient to carry, had less of an unpleasant taste, and caused less nasal irritation than the aqua spray. More than twice as many patients preferred Turbuhaler to the aqua spray (69% versus 31%). Improvement in quality of life from baseline to clinic visits was statistically significant in both groups. CONCLUSION Once daily use of 256 mg of budesonide aqua and 400 mg of budesonide Turbuhaler are equally safe and efficacious in the treatment of SAR. Patients preferred the budesonide powder formulation delivered via Turbuhaler two to one over the aqua formulation.

Ground truth y :

P: "seasonal allergic rhinitis .", "Two hundred and eighty-four", "out-patients", "SAR", "symptomatic during the ragweed season", "243", "SAR ."

I: "budesonide Turbuhaler", "budesonide aqua", "budesonide .", "Turbuhaler", "aqua spray", "budesonide powder", "aqua"

C: (none)

O: "patients ' preferences", "quality of life", ".", "Mean daily nasal symptom scores", "eye symptoms .", "nasal and non-nasal adverse events", "epistaxis", "headache .", "nasal irritation", "safe", "efficacious"

Candidate A:

P: "out-patients with SAR" , "seasonal allergic rhinitis", "ragweed season" , ~~"Two-hundred-and-eighty-four"~~ , ~~"out-patients"~~ , ~~"SAR"~~ , ~~"symptomatic during the ragweed season"~~ , ~~"243"~~ , ~~"SAR."~~

I: "budesonide Turbuhaler 400 microg/day" , "budesonide powder formulation" , "Turbuhaler", ~~"budesonide Turbuhaler"~~ , ~~"budesonide aqua"~~ , ~~"budesonide."~~ , ~~"aqua spray"~~ , ~~"budesonide powder"~~ , ~~"aqua"~~

C: "budesonide aqua 256 microg/day" , "aqua spray" , "aqua formulation"

O: "nasal symptom scores" , "eye symptoms", "control of symptoms" , "adverse events" , "epistaxis", "headache" , "ease of use" , "convenient to carry" , "unpleasant taste" , "nasal irritation", "patients' preferences", "quality of life", "safe", "efficacious", ~~"."~~ , ~~"Mean daily nasal symptom scores"~~ , ~~"nasal and non-nasal adverse events"~~

Candidate B:

P: "out-patients with SAR" , "seasonal allergic rhinitis", "ragweed season" , ~~"Two-hundred-and-eighty-four"~~ , ~~"out-patients"~~ , ~~"SAR"~~ , ~~"symptomatic during the ragweed season"~~ , ~~"243"~~ , ~~"SAR."~~

I: "budesonide Turbuhaler 400 microg/day" , "budesonide powder formulation" , "Turbuhaler device" , ~~"budesonide Turbuhaler"~~ , ~~"budesonide aqua"~~ , ~~"budesonide."~~ , ~~"Turbuhaler"~~ , ~~"aqua spray"~~ , ~~"budesonide powder"~~ , ~~"aqua"~~

C: "budesonide aqua 256 microg/day" , "aqua spray" , "aqua formulation"

O: "nasal symptom scores" , "eye symptoms", "control of symptoms" , "adverse events" , "epistaxis", "headache" , "ease of use" , "convenience" , "taste" , "nasal irritation", "patients' preferences", "quality of life", "safe", "efficacious", ~~"."~~ , ~~"Mean daily nasal symptom scores"~~ , ~~"nasal and non-nasal adverse events"~~

Scoring with macro-F1 exact match m against y :

Macro-F1 exact match lowercases spans and strips punctuation before testing equality, then scores set overlap per PICO key. For each key $k \in \{P, I, C, O\}$, treat $\mathcal{Y}_k$ and the candidate spans $\mathcal{C}_k$ as sets; count TP, FP, and FN, then $P_k = \text{TP}/(\text{TP} + \text{FP})$, $R_k = \text{TP}/(\text{TP} + \text{FN})$, and $F1_k = 2P_kR_k/(P_k + R_k)$.

Candidate A:

- P : TP=1, FP=2, FN=6 $\Rightarrow P=0.33, R=0.14, F1=0.20$
- I : TP=1, FP=2, FN=6 $\Rightarrow P=0.33, R=0.14, F1=0.20$
- C : TP=0, FP=3, FN=0 $\Rightarrow P=0.00, R=0.00, F1=0.00$
- O : TP=8, FP=6, FN=3 $\Rightarrow P=0.57, R=0.73, F1=0.64$

$$m(A, y) = \frac{1}{4} \sum_k F1_k = 0.26.$$

Candidate B:

- P : TP=1, FP=2, FN=6 $\Rightarrow P=0.33, R=0.14, F1=0.20$
- I : TP=0, FP=3, FN=7 $\Rightarrow P=0.00, R=0.00, F1=0.00$
- C : TP=0, FP=3, FN=0 $\Rightarrow P=0.00, R=0.00, F1=0.00$

- O : TP=8, FP=6, FN=3 $\Rightarrow P=0.57, R=0.73, F1=0.64$

$$m(B, y) = 0.21.$$

Pair:

Chosen: A ($s = 0.26$); Rejected: B ($s = 0.21$).

A.2 MedCalc

Split=test, ID=8.

Source text x :

A 71-year-old man, non-smoker (height: 155 cm; weight: 59 kg), was scheduled to receive video-assisted thoracoscopic extended thymectomy with the diagnosis of MG. Two months previously, he developed symptoms of right ptosis and progressive swallowing difficulty. Based on a positive response to edrophonium and increased titers of autoantibodies to acetylcholine receptor (19.3 nmol/L; normal < 0.2 nmol/L), he was diagnosed as having MG with severity belonging to Osserman's classification IIb (ie, generalized moderate weakness and/or bulbar dysfunction). Thoracic computed tomography demonstrated glandular hyperplasia of the thymus (Fig. A). The patient was started on prednisolone 20 mg daily and pyridostigmine 60 mg three times daily. His past history included hypertension without evidence of previous myasthenic crisis or thromboembolic events (eg, history of lower limb swelling). The results of electrocardiography, pulmonary function test [eg, vital capacity: 93%], echocardiography (eg, left ventricular ejection fraction: 85.1%), chest radiography (Fig. B), and laboratory studies (eg, coagulation test) were unremarkable. On the other hand, impaired renal function [i.e., serum creatinine: 1.42 mg/dL; eGFR: 49.1 mL/min/1.73 m²] was observed after admission. Preoperative physical examination of the patient showed clear consciousness without respiratory distress. Vital signs included a blood pressure of 187/103 mm Hg, heart rate of 82 beats/min, and respiratory rate of 14 breaths/minute. Under real-time neuromuscular monitoring with a train-of-four (TOF) monitor (TOF-watch SX, N. V. Organon, Oss, Netherlands), anesthesia was induced with propofol (130 mg) and rocuronium (0.85 mg/kg). Following successful tracheal intubation with a double-lumen tracheal tube (Broncho-Cath; Mallinckrodt, Athlone, Ireland), general anesthesia was maintained with sevoflurane, rocuronium (total dosage: 40 mg), and a continuous infusion of remifentanyl. An 18-gauge peripheral intravenous line and an arterial line were introduced. The surgical time was 4 hours 15 minutes with an estimated blood loss of 100 mL. Upon completion of surgery, sugammadex 4 mg/kg was administered to reverse neuromuscular blockade, with a maximum TOF ratio of 0.93 following reversal. Additionally, intravenous morphine 8 mg was given for postoperative analgesia. After successful extubation in the operating room and resumption of spontaneous breathing, he was transferred to the post-anesthesia care unit (PACU) for further care. During the immediate postoperative period, he was hemodynamically stable without respiratory distress. Because of surgical pain with a numeric rating scale of 5 (scale of 0-10), intravenous morphine was titrated to a total dosage of 7 mg. Forty-five minutes later, respiratory distress with drowsiness was noted. Physical examination found pinpoint pupils with a TOF ratio of 0.9. Blood gas analysis demonstrated severe hypercapnia (arterial carbon dioxide pressure: 117.7 mm Hg) and acidosis (pH: 6.996, lactate levels: 3.3 mmol/L). On suspicion of morphine overdose, intravenous naloxone was administered twice (0.08 mg each time). After 20 minutes, the patient regained consciousness and normal respiratory pattern. Subsequent blood gas analysis demonstrated hyperlactatemia (lactate levels: 6.0 mmol/L) (Fig.) despite improvement in arterial carbon dioxide pressure and arterial oxygen pressure after naloxone administration. Taking into account the overall clinical improvement, he was transferred to ward with spontaneous breathing and stable hemodynamics after observation for 100 minutes following initial reversal of opioid overdose. However, one hour after being transferred to ward (ie, three hours after naloxone administration), he was found to exhibit consciousness loss, respiratory distress, and pinpoint pupils. Intravenous naloxone 0.4 mg was given, followed by endotracheal intubation and transfer to the intensive care unit for mechanical ventilatory support. Brain computed tomography showed no intracranial

lesion. He regained consciousness two hours after naloxone administration in the intensive care unit without symptoms of opioid withdrawal (eg, pulmonary edema). As weaning from mechanical ventilation was difficult on POD 3, a diagnosis of POMC was made. Steroid therapy (prednisolone 40 mg twice daily) and pyridostigmine (60 mg 3 times a day) was initiated. The patient was extubated smoothly on POD 6 and was discharged from hospital on POD 12. The course of hyperlactatemia was shown in Figure. In addition, his eGFR increased from 49.1 mL/min/1.73m² at baseline to 88.9 mL/min/1.73m² on POD 5. Pathological analysis of the specimen from thymectomy confirmed the diagnosis of type B2 thymoma according to the World Health Organization (WHO) classification. The patient was readmitted on POD 24 because of left thigh swelling. Ultrasonographic examination showed evidence of DVT involving the left femoral and popliteal veins. Anticoagulant therapy with low-molecular-weight heparin (ie, subcutaneous Clexane 60 mg every 12 hours) was implemented immediately after hospitalization, and he was discharged without sequelae on POD 31. During hospitalization, his eGFR was 86.5 mL/min/1.73m². There was no recurrence of myasthenic crisis or DVT up to 3 months of follow-ups. What is the patient's Creatinine Clearance using the Cockcroft-Gault Equation in terms of mL/min? You should use the patient's adjusted body weight in kg instead of the patient's actual body weight if the patient is overweight or obese based on their BMI. If the patient's BMI is normal, set their adjusted body weight to the minimum of the ideal body and actual weight. If the patient is underweight, please set their adjusted body weight to their actual body weight. You should use the patient's medical values and health status when they were first admitted to the hospital prior to any treatment.

Ground truth y :

```
age: 71
sex: male
adjusted_body_weight_kg: 52.35
serum_creatinine_mg_dL: 1.42
clearance_mL_min: 35.33
```

Candidate A:

```
age: 71
sex: male
adjusted_body_weight_kg: 52.35 52.7
serum_creatinine_mg_dL: 1.42
```

Candidate B:

```
age: 71
sex: male
adjusted_body_weight_kg: 52.35
serum_creatinine_mg_dL: 1.42
```

Scoring with MSE m against y :

Cockcroft-Gault:

```
CrCl(age, sex, adjusted_body_weight_kg, serum_creatinine_mg_dL) =
((140 - age) * adjusted_body_weight_kg * (1 if sex == 'male' else 0.85))
/ (serum_creatinine_mg_dL * 72)
```

$m(c, y) = \text{mean}((\text{CrCl}(c) - y)^2)$, with a single scalar clearance target.

Candidate A:

```
CrCl(71, "male", 52.7, 1.42)
= ((140 - 71) * 52.7 * 1) / (1.42 * 72)
= 35.57
```

$$m(A, y) = \text{mean}((35.57 - 35.33)^2) = \text{mean}(0.23^2) = 0.054.$$

Candidate B :

```
CrCl(71, "male", 52.35, 1.42)
= ((140 - 71) * 52.35 * 1) / (1.42 * 72)
= 35.33
```

$$m(B, y) = \text{mean}((35.33 - 35.33)^2) = 0.0.$$

Pair:

Chosen: B ($s = 0.0$); Rejected: A ($s = 0.054$).

A.3 MedBullets

Split=MedBullets/train, ID=0.

Source text x :

A 64-year-old man presents to the emergency room with a headache and nausea. He reports that he was rocking his grandson to sleep when the symptoms began. He states the pain is constant and is primarily located on his right side. When asked to indicate the area of pain, he says that it surrounds his eye and upper forehead. He had one episode of vomiting. The patient also reports difficulty seeing out of his right eye, which he attributes to excessive tearing. The patient's past medical history is significant for hypertension. His medications include hydrochlorothiazide. His temperature is 98.6°F (37°C), blood pressure is 135/91 mmHg, pulse is 72/min, and respirations are 12/min. The patient's right eye is shown in Figure A. Upon physical examination, the right pupil is minimally responsive to light and the globe feels firm. A right-sided carotid bruit is appreciated. Which of the following is the most appropriate prophylaxis for this patient's condition? Options: A. Acetazolamide B. Amitriptyline C. Clopidogrel D. Epinephrine E. Verapamil

Ground truth y :

A. Acetazolamide

Candidate A :

A. Acetazolamide

Candidate B :

~~A. Acetazolamide~~ C. Clopidogrel

Scoring with boolean correctness m against y :

$$m(c, y) = 1.0 \text{ if } c \text{ exactly matches } y, \text{ else } 0.0.$$

$$\text{Candidate } A: m(A, y) = 1.0 \text{ if } c=y \text{ else } 0.0 = 1.0.$$

$$\text{Candidate } B: m(B, y) = 1.0 \text{ if } c=y \text{ else } 0.0 = 0.0.$$

Pair:

Chosen: A ($s = 1.0$); Rejected: B ($s = 0.0$).

B Prompt templates

The prompts below are adapted from the J1 [10] thinking seed templates. We reuse them throughout the pipeline: when generating distillation data for SFT, when tuning with SFT or RL, and finally when meta-evaluating. Every judge prompt has a system message:

```
You are an impartial judge evaluating how well response(s) fulfill the
user's instructions.
```

and a user message composed conceptually of four blocks: 1) *format instructions*, 2) *task instructions* (also called “rubric”), 3) *source text*, and 4) *candidate response(s)*. The exact contents of the user message template vary by judge mode, since we need different format instructions for pairwise or pointwise, and by task, as task instructions are idiosyncratic. We detail each below.

B.1 Pairwise

```
You are given a user question and two responses from two AI assistants.
Your task is to act as an impartial judge and evaluate which response
better follows the user's instructions and provides a higher-quality
answer.
```

```
Think carefully about how to assess the quality of the responses, and
enclose your reasoning within <think> and </think> tags. Your reasoning
should include your evaluation criteria, a clear understanding of what
an ideal response would look like for this particular question, and a
concrete example of such an ideal or reference answer if possible. Then
compare both assistants' responses to your ideal or reference answer,
explaining how each aligns with or deviates from your expectations. Be
specific and avoid vague or overly general judgments. Remain as
objective as possible.
```

```
Finally, provide your verdict within <answer> and </answer> tags,
strictly following this format:
```

```
- <answer> [[A]] </answer> if Assistant A is better
- <answer> [[B]] </answer> if Assistant B is better
```

```
Format your output like this:
```

```
<think> your_thinking_process </think>
<answer> [[A]] </answer> (or [[B]])
```

```
Below are the user's question and the two responses:
```

```
[User Question]
{instruction}
{text}
```

```
[The Start of Assistant A's Answer]
{response_A}
[The End of Assistant A's Answer]
```

```
[The Start of Assistant B's Answer]
{response_B}
[The End of Assistant B's Answer]
```

where:

- {instruction} — task instructions (or “rubric”) from §B.3 (PICO, MedCalc, or QA).

- `{text}` — source passage for the example (e.g., abstract, clinical note, or exam stem), from the task benchmark via §2.2.
- `{response_A}` — candidate output in slot *A* of a preference pair, from the mutation operator in §2.2 (worked examples in Appendix A); slot assignment varies across training batches to mitigate position bias.
- `{response_B}` — candidate output in slot *B*, same provenance as `{response_A}`.

B.2 Pointwise

You are given a user question and a response from an AI assistant. Your task is to act as an impartial judge and evaluate how well the response fulfills the user's instructions.

Think carefully about how to assess the quality of the response, and enclose your reasoning within `<think>` and `</think>` tags. Your reasoning should include your evaluation criteria, a clear understanding of what an ideal response would look like for this particular question, and a concrete example of such an ideal or reference answer if possible. Then compare the assistant's response to your ideal or reference answer, explaining how it aligns with or deviates from your expectations. Be specific and avoid vague or overly general judgments. Remain as objective as possible.

Finally, assign the assistant's response a score from 0 to 10, using either an integer or a decimal with up to 0.1 precision. A higher score should indicate a higher-quality response. Enclose the score within `<score>` and `</score>` tags.

Format your output like this:
`<think> your_thinking_process </think>`
`<score> your_score </score>`

Below are the user's question and the assistant's response:

[User Question]
`{instruction}`
`{text}`

[The Start of the Assistant's Answer]
`{response}`
 [The End of the Assistant's Answer]

where:

- `{instruction}` — task instructions (or “rubric”) from §B.3 (PICO, MedCalc, or QA).
- `{text}` — source passage for the example, same provenance as in the pairwise template.
- `{response}` — a single candidate output from §2.2, presented one at a time for pointwise judging (or in distillation rows).

B.3 Task instructions

PICO

Extract PICO elements from this biomedical text. Identify Population (P), Intervention (I), Comparison (C), and Outcome (O) components mentioned in the study. Output should be a JSON object with keys "P", "I", "C", and "O", where values are arrays of strings. Each string should be a verbatim copy of the relevant text from the original document. Keep each string as short as possible, containing only the topic words. Avoid including all connectors and other words that are not part of the topic. Numbers should be separated from the following words they refer to.

MedCalc

You are given the generic formula, together with a dictionary containing the definitions of all variables. Find in the patient note the current values for these variables and substitute them in the formula, do measurements conversion if needed. All variables have to be specified. Also manipulate the formula to make it Python executable in a single line, with no imports or intermediate variables definitions. Do not try to calculate the results of the formula, just write the correct values of the variables. If working with dates or time deltas, use the Python library datetime which has been already imported as from datetime import datetime, timedelta. For mathematical functions, such as log, use the Python library numpy which has been already imported as import numpy as np. If a specific formatting is required, unless it is a simple number, specify it as if it had to be executed as a Python string. Make sure that all the specific variables needed to compute accurately the formula have been provided, otherwise signal it in the all_inputs flag. If the value of a categorical boolean variable can not be determined from the description, for instance something like Diabetes_history, assume it to be False and do not flag it as absent in the all_inputs flag. If the value of another kind of variable can not be determined, check how it is used in the formula. If it is used only in a comparison, such as age < 65, assume it to be a value that makes the comparison false. The selected formula {calculator} has the following definition: {formula}. The variables are defined as: {formula_inputs}. The output format has to be {out_format}.

where:

- {calculator} — name of the selected MedCalc-Bench formula for this example (from the benchmark metadata).
- {formula} — mathematical definition of that formula (from the benchmark's formula catalog).
- {formula_inputs} — variable definitions and units required by the formula.
- {out_format} — required output format for the substituted, Python-evaluable formula string.

QA (MedQA, MedBullets, PubMedQA)

You are given a medical question and a set of possible answers. Select the most appropriate answer from the given options.

C Detailed results

Figure 1 summarizes pairwise judging correctness, robustness, and compliance; Tables 2, 3, and 4 report the exact per-suite values. Figure 2 summarizes rank-gap difficulty on decomposable items; Table 5 reports the exact per-tertile values.

	base	SFT	RL	SFT→RL	O4-mini	GPT-5	Opus 4.6
Decomposable ($n=1026$)	0.75	0.79* (+0.04)	0.67* (−0.08)	0.84* (+0.09)	0.82	0.87	0.86
PICO ($n=187$)	0.84	0.88 (+0.04)	0.78 (−0.06)	0.94* (+0.10)	0.86	0.94	0.94
MedCalc ($n=839$)	0.65	0.71* (+0.06)	0.57* (−0.08)	0.75* (+0.10)	0.79	0.80	0.79
Atomic ($n=2081$)	0.76	0.79* (+0.03)	0.69* (−0.07)	0.78 (+0.02)	0.91	0.91	0.91
MedBullets ($n=308$)	0.69	0.72 (+0.03)	0.63* (−0.06)	0.73 (+0.04)	0.94	0.93	0.94
MedQA ($n=1273$)	0.79	0.82* (+0.03)	0.71* (−0.08)	0.78 (−0.01)	0.96	0.97	0.96
PubMedQA ($n=500$)	0.79	0.83* (+0.04)	0.74* (−0.05)	0.84* (+0.05)	0.83	0.84	0.84
Avg	0.75	0.79* (+0.04)	0.69* (−0.06)	0.81* (+0.06)	0.87	0.89	0.89

Table 2: Pairwise judging **correctness** with rows labeled **Decomposable** and **Atomic** as unweighted family means, indented rows as individual benchmarks, **Avg** as the unweighted mean across all five suites, **Bold** as best-in-class among Llama-8B or frontier columns, underlined bold as overall row best, and asterisk as McNemar significance [20] vs. baseline ($p < 0.05$).

	base	SFT	RL	SFT→RL	O4-mini	GPT-5	Opus 4.6
PICO ($n=187$)	0.14	0.06 (−0.08)	0.03 (−0.11)	0.06 (−0.08)	0.09	0.03	0.02
MedCalc ($n=839$)	0.07	0.05 (−0.02)	0.03 (−0.04)	0.02 (−0.05)	0.10	0.01	0.02
MedBullets ($n=308$)	0.05	0.04 (−0.01)	0.06 (+0.01)	0.07 (+0.02)	0.11	0.01	0.01
MedQA ($n=1273$)	0.04	0.09 (+0.05)	0.09 (+0.05)	0.04 (+0.00)	0.07	0.01	0.00
PubMedQA ($n=500$)	0.08	0.06 (−0.03)	0.11 (+0.03)	0.06 (−0.02)	0.03	0.01	0.00
Avg	0.08	0.06 (−0.02)	0.07 (−0.01)	0.05 (−0.03)	0.08	0.01	0.01

Table 3: **Verdict variance** is the mean $|\text{high} - \text{low}|$ across datasets over majority-vote and any-/all-sample correctness rates (lower is better), with the same columns and formatting as Table 2.

	base	SFT	RL	SFT→RL	O4-mini	GPT-5	Opus 4.6
PICO ($n=187$)	0.0%	4.5% (+4.5)	0.0% (+0.0)	0.0% (+0.0)	3.7%	0.0%	0.0%
MedCalc ($n=839$)	5.5%	6.5% (+1.0)	4.7% (−0.8)	4.2% (−1.3)	0.0%	0.0%	0.0%
MedBullets ($n=308$)	1.1%	0.0% (−1.1)	4.3% (+3.2)	0.0% (−1.1)	0.0%	0.0%	0.0%
MedQA ($n=1273$)	0.4%	0.9% (+0.5)	3.8% (+3.4)	0.7% (+0.3)	12.8%	0.0%	0.0%
PubMedQA ($n=500$)	0.0%	0.0% (+0.0)	0.0% (+0.0)	0.0% (+0.0)	1.2%	0.0%	0.0%
Avg	2.3%	2.8% (+0.5)	3.6% (+1.3)	1.6% (−0.7)	2.2%	0.0%	0.0%

Table 4: **Invalid-response share** is invalid / (invalid + legitimate errors), lower is better, with the same columns and formatting as Table 2.

	base	SFT	RL	SFT→RL	O4-mini	GPT-5	Opus 4.6
Hard ($n=439$)	0.62	0.65 (+0.03)	0.54* (−0.08)	0.69* (+0.07)	0.69	0.72	0.72
Medium ($n=440$)	0.72	0.79* (+0.07)	0.63* (−0.09)	0.83* (+0.11)	0.86	0.88	0.87
Easy ($n=147$)	0.79	0.86 (+0.07)	0.72 (−0.07)	0.90* (+0.11)	0.94	0.94	0.95

Table 5: **Rank-gap difficulty** on decomposable items (PICO + MedCalc), where **Hard** / **Medium** / **Easy** are three tertiles of δ per source example with boundaries over $[1, K-1]$ that depend on that example’s candidate count K (hard = smallest gaps), not fixed absolute integers across tasks, using the same columns and formatting as Table 2.