MAGMA: MULTIMODAL AGENTIC MEDICAL AI WITH CONTEXT-BASED ITERATIVE REFINEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Predictive machine learning models hold substantial promise in healthcare, yet their development typically requires significant domain expertise and careful methodological reporting. Recent agentic AI systems have demonstrated strong coding capabilities, including for end-to-end machine learning pipeline construction. However, their use in healthcare raises additional requirements since workflows must be clinically-grounded, transparent, and reported in sufficient detail to support appraisal and reproducibility. We introduce MAGMA (Multimodal AGentic Medical AI), an agentic system for end-to-end clinical model development and evaluation. MAGMA's central contribution is a judge-driven iterative refinement loop in which a judge agent audits each attempt against a rubric derived from established clinical prediction modelling guidelines and returns a composite refinement score combining predictive performance with reporting compliance. This score and the item-level feedback guide the execution of the next attempt, so that clinical reporting compliance drives refinement. We also constructed MAGMA-Suite, a new benchmark of 10 clinical case studies spanning structured electronic health records, CXR images, ECG signals, and clinical text. We also propose MAGMAScore, a guideline-derived 22-item metric for assessing the clinical grounding of autonomous modeling workflows. All systems are evaluated on a fixed, patient-disjoint held-out partition across four prompt templates and three independent runs per configuration, and results are reported on a completion-weighted basis that accounts for failed runs. Under this protocol, single-agent baselines complete only 22.5% of attempted runs, while MAGMA completes every attempted run, and attains the highest completion-weighted AUROC in 9 of the 10 case studies. In a blinded expert adjudication conducted by two clinical machine-learning assessors against a rubric held out from the refinement loop, MAGMA received the highest score in 8 of 10 case studies (mean 0.74 versus 0.62 across baseline packages). We release the system, benchmark, and evaluation pipelines to support reproducible research on autonomous machine learning in healthcare.

1 INTRODUCTION

Machine Learning (ML) and Artificial Intelligence (AI) have substantial potential to transform healthcare processes by supporting diagnosis, clinical decision-making, and patient care (Meng et al., 2024). By learning patterns across patient populations, ML models can help apply clinical knowledge consistently and at scale, complementing individual clinician judgment with reproducible data-driven inference (Shamout et al., 2020; Feng et al., 2026). However, developing ML models for clinical applications remains challenging. It requires an understanding of the underlying medical context and clinical workflows, as well as technical expertise to design, train, validate, and report models end-to-end (de Almeida et al., 2025).

To promote transparent and reproducible development and evaluation of AI models in healthcare, the clinical research community has established reporting guidelines for clinical prediction models. Among the most widely used is TRIPOD+AI (Collins et al., 2024), which provides reporting guidance across key aspects of study design, model development, validation, and intended use. Automated machine learning (AutoML) systems offer a pathway toward autonomous model development, including both general-purpose systems (Fang et al., 2025) and healthcare-focused platforms (Zhu et al., 2025; Saveliev et al., 2025). However, existing AutoML approaches are not explicitly aligned with

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

established clinical reporting guidelines and typically optimize primarily for predictive performance (Thirunavukarasu et al., 2023). Moreover, they are often constrained to a single data modality, despite the rapidly growing efforts in multimodal learning for healthcare applications.

To address this gap, we propose MAGMA (Multimodal AGentic Medical AI), a clinically grounded multi-agent system for end-to-end clinical prediction model development from natural language instructions. MAGMA autonomously constructs and executes machine learning pipelines while incorporating clinical guidelines and expert input throughout the workflow. Rather than treating reporting guidelines as a post-hoc checklist, MAGMA operationalizes clinical reporting guidance as both a set of constraints during pipeline construction and a refinement signal for iterative model improvement and selection. To evaluate MAGMA and relevant baselines, we introduce MAGMA-Suite, a benchmark of 10 unimodal and multimodal clinical case studies spanning four modalities and seven source datasets. The benchmark covers a diverse set of clinically relevant prediction tasks and is designed to evaluate the full autonomous model development pipeline, from task specification and data processing to model training, validation, and reporting. We further introduce MAGMAScore, an automated 22-item metric derived principally from TRIPOD+AI that quantifies the extent to which an autonomous ML system produces clinically grounded and guideline-compliant workflows. Together with a blinded expert adjudication evaluation framework scored on a separate, external rubric, this enables systematic comparison of MAGMA against existing general-purpose and multi-agent baselines under a fixed held-out protocol. In summary, we make the following contributions:

1. We propose **MAGMA**, a multimodal agentic system for end-to-end clinical model development and evaluation, which includes a clinically-grounded refinement-score formulation.
2. We introduce **MAGMA-Suite**, a benchmark suite of 10 clinical case studies spanning four modalities and seven source datasets, designed to evaluate autonomous ML systems on clinically relevant unimodal and multimodal prediction tasks.
3. We propose a new evaluation framework comprising **MAGMAScore**, an automated 22-item rubric derived principally from TRIPOD+AI for assessing guideline compliance and clinical grounding, together with a separate eight-item rubric scored by two blinded expert assessors for holistic workflow evaluation.
4. We release the system, benchmark, rubric, and evaluation pipelines to support reproducible research on clinically grounded autonomous machine learning.

2 Related Work

2.1 General-Purpose Automated ML Systems

Recently, several AutoML systems emerged for end-to-end pipeline development and evaluation, achieving competitive performance across a wide range of tasks and domains with minimal human intervention (Table 1). For example, FLAML (Wang et al., 2021) and H2O AutoML (H2O.ai, 2026) focus on algorithm selection and hyperparameter tuning. AutoGluon (Erickson et al., 2020) replaces single-model selection with multi-layer stacking and automatic feature engineering, allowing it to consistently achieve strong performance on structured benchmarks. MLZero introduces an agentic architecture, where the pipeline is decomposed into specialized agents responsible for data understanding, feature engineering, and model training, achieving better results than its previous counterparts (Fang et al., 2025). While effective in their scope, these systems share the fundamental limitation of being designed to achieve maximum performance on arbitrary data with no awareness of the domain-specific constraints that govern such applications. In other words, they are designed to accept any tabular input and optimize any scalar metric, with no notion of outcome definition, cohort validity, or data cleanliness. Furthermore, pipeline construction in said systems is static and preset, with no mechanism to incorporate human-in-the-loop (HIL).

2.2 Medical-Centric Frameworks

Recent work specifically targets automated clinical prediction modeling (Gao et al., 2025). For example, HACHI (Feng et al., 2026) follows a human-AI co-design approach that structures the pipeline as two nested loops. However, it is constrained to textual notes, its inner loop issues multiple LLM API

calls per concept evaluation cycle, making end-to-end runs computationally expensive at scale, and the linear model architecture limits predictive capacity on tasks requiring nonlinear feature interactions. HealthFlow (Zhu et al., 2025) presents a self-evolving agentic system including a "reflector" agent that allows for strategic learning by storing successes and failures from past executions into a structured experience base, and a "meta-planning" agent that improves future strategies. The CliMB-DC (Saveliev et al., 2025) architecture combines a "coordinator" agent that generates an end-to-end execution plan and incorporates user feedback at critical decision points via a HIL mechanism, with a "worker" agent that generates code and calls tools to execute the generated plan. However, the system is unimodal and constrained to tabular data only, with no support for other modalities, and it is not grounded by clinical reporting guidelines. MAGMA addresses these gaps through two design specifications needed for responsible and effective clinical use: (i) native support for multimodal clinical prediction tasks, spanning structured EHR, clinical text, chest X-rays, ECG signals, and their combinations; and (ii) explicit compliance enforcement and guideline-driven iterative refinement based on TRIPOD+AI reporting guidelines.

Table 1: Comparison of MAGMA with existing AutoML systems across dimensions relevant to clinical prediction model development.

System	Medical	Modalities	Agentic	HIL	Clinically Grounded
FLAML (Wang et al., 2021)	×	Tabular	×	×	×
H2O AutoML (H2O.ai, 2026)	×	Tabular	×	×	×
AutoGluon (Erickson et al., 2020)	×	Tabular	×	×	×
MLZero (Fang et al., 2025)	×	Tabular	✓	×	×
HACHI (Feng et al., 2026)	✓	Text	×	✓	×
HealthFlow (Zhu et al., 2025)	✓	Tabular	✓	×	×
MoMA (Gao et al., 2025)	✓	Tabular + Img + Text	✓	×	×
CliMB-DC (Saveliev et al., 2025)	✓	Tabular	✓	✓	×
MAGMA (Ours)	✓	**Tabular + Text + Img + ECG**	✓	✓	✓

3 METHODOLOGY

3.1 MAGMA ARCHITECTURE

3.1.1 SYSTEM FORMULATION & AGENTIC ARCHITECTURE

We formulate MAGMA as a finite-horizon, orchestrator-guided iterative refinement process. The full workflow is shown in Figure 1. The workflow includes eight actors, an orchestrator responsible for managing the workflow, calling agents, interacting with the user, and managing retries as needed, and a set of specialized actors:

$$\mathcal{W} = \{A_D, A_{Dv}, A_M, A_{Mv}, A_J, A_H\}.$$

A_D is the data worker responsible for data preprocessing and cleaning, and target data derivation, A_{Dv} is the data validator responsible for validating the data worker's executed code, A_M is the model worker responsible for model training, A_{Mv} is the model validator responsible for validating the model worker's executed code, A_J is the judge agent responsible for assessing the entire workflow and computes the composite refinement score used by the orchestrator, and A_H represents the human-in-the-loop instructions.

Each attempt by the orchestrator t begins with an execution context c_t holding the task specification, the artifacts produced by prior attempts, and any retry instructions. The current shared state s_t is also maintained. The orchestrator selects an actor $w_t \in \mathcal{W}$, which is invoked as

$$y_t = w_t(s_t, c_t),$$

where y_t is the produced output. The shared state is then updated as

$$s_{t+1} = U(s_t, y_t).$$

In our implementation, U corresponds to writing new artifacts to the run directory, including handoff JSON files, validation results, judge's feedback, and retry instructions. Within an attempt, the data and model workers generate artifacts that their validators either accept or return for repair, and the accepted artifacts are evaluated by the judge.

3.1.2 JUDGE-DRIVEN ITERATIVE REFINEMENT

The judge agent acts as an automated auditor of each attempt, using an iterative refinement loop. After every attempt, the judge assesses the execution against the 22 TRIPOD+AI-derived criteria, calculates

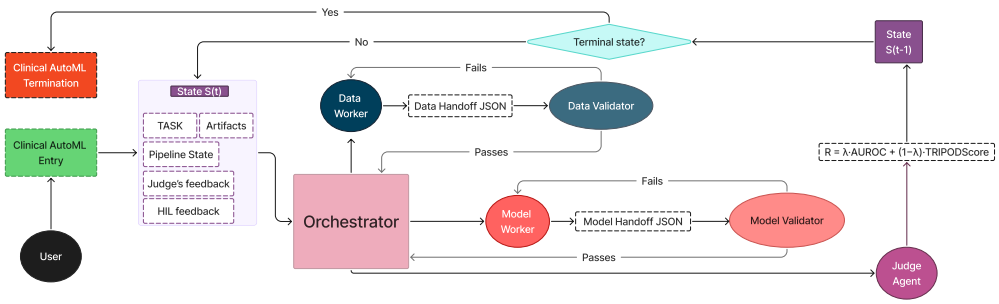


Figure 1: **MAGMA Architecture Overview.** MAGMA uses an orchestrator to coordinate a data worker, data validator, model worker, model validator, human-in-the-loop feedback, and judge agent for end-to-end clinical model development. The judge agent audits the run using the MAGMAScore rubric, and the resulting composite refinement score considers predictive performance and reporting compliance, conditioning the execution context of the next attempt until the system reaches a valid model, its attempt limit, or a plateau.

the MAGMAScore (Section 3.2) using Yes or No verdicts, and writes both: (i) a judge verdict JSON file with item-level scores and identified failures; and (ii) a score file containing the calculated scalar refinement score to be used by the orchestrator. The mechanism provides actionable item-level verdicts: each unsatisfied item maps to a specific step in the end-to-end pipeline, so a failing run can be localised and repaired rather than merely labelled deficient. The composite refinement score combines predictive performance, measured by the Area Under the Receiver Operating characteristic Curve (AUROC), with clinical alignment, measured by MAGMAScore:

$$S_t = \lambda \cdot \text{AUROC}_t + (1 - \lambda) \cdot \text{MAGMAScore}_t, \tag{1}$$

where AUROC is computed by the judge from the model worker's predictions on an internal validation split, and MAGMAScore is the proportion of items satisfied. The internal validation split is carved out of the exposed training partition by the data worker as part of the pipeline it constructs, and it is the only split available to the system during refinement. The held-out partition is never exposed to any system at any point in the loop, and is used exclusively after a run terminates, to score the final selected model. Attempt selection and termination therefore depend only on internal validation performance.

The orchestrator continues to call the worker agents for as long as the system has not reached a terminal state. An attempt sequence terminates when S_t reaches the target η value of a relevant metric, when the attempt budget is exhausted, or when S_t fails to improve by at least ϵ over a plateau window of p attempts. Appendix A states the update, validator, and termination rules formally. We note explicitly that no model parameters are updated and nothing is carried from one attempt sequence to the next: refinement operates entirely through the execution context.

3.1.3 Human-in-the-Loop Clarification

While most AutoML systems, including MAGMA, are designed to operate autonomously once a task is specified, real-world clinical AutoML requests are often ambiguous or missing crucial information, and hallucinating missing information is unsafe, especially in clinical settings. As such, we introduce two HIL entry points in MAGMA: (i) before the pipeline starts, the orchestrator inspects the prompt and decides whether the message contains sufficient information or if further clarifications are needed. If the latter option is true, the orchestrator generates a single follow-up question, and the user's answer is appended to the run state; and (ii) during execution, any agent can send a human input request when it encounters ambiguity that it cannot resolve. This also leads the orchestrator to generate a single follow-up question, and the user's answer is appended to the run state. This design allows us to maintain the autonomous operation of MAGMA without needing to ask for user clarification at every step, while also ensuring that genuinely missing information can be resolved instead of hallucinating.

3.2 MAGMASCORE: GUIDELINE-GROUNDED REFINEMENT SIGNAL

In ML development, multiple metrics are typically reported to capture a model's performance on a given task since, generally, no single metric is capable of capturing all the desirable properties of said model (Hicks et al., 2021). In the case of clinical AI specifically, the integration of AI tools into medical workflows requires an extra level of robustness and consistency to ensure reliability, clinical relevance, and patient safety (Kopanichuk et al., 2025), and traditional metrics often fail to capture human-centered concerns, especially in imbalanced settings (Cabitza, 2026). We therefore introduce MAGMAScore, a 22-item score that captures the clinical alignment of an autonomous modelling workflow through its compliance with established reporting standards for clinical prediction models.

The items are drawn principally from TRIPOD+AI (Collins et al., 2024), the most widely adopted reporting guideline for AI-based clinical prediction models, and supplemented where that guideline is silent on properties specific to automated or LLM-driven pipelines: TRIPOD-LLM (Gallifant et al., 2024) for compute and cost reporting and for consistency of preprocessing across sociodemographic groups, MI-CLAIM-GEN (Miao et al., 2024) for patient-level partitioning, and the functional-requirements framework for clinical AutoML platforms (Scott et al., 2024) for capabilities such as feature engineering, model comparison, and hyperparameter search. We retained only items whose satisfaction can be judged automatically against the artifacts an autonomous system can be expected to produce, namely the model, its training code, its evaluation outputs, and any accompanying documentation. Items requiring artifacts outside the scope of an automated pipeline, such as IRB approval, dissemination plans, and stakeholder consultation, were excluded. The resulting 22 items span five categories: problem framing, data preparation, analytical methods, model specification, and model performance and evaluation. Appendix B lists them in full with their source mappings.

Each item is scored as Yes or No based on whether the system's output satisfies the reporting requirement. A Yes indicates the item is fully addressed, and a No indicates it is missing or insufficient. MAGMAScore for a given run is computed as

$$\text{MAGMAScore} = \frac{\text{items scored Yes}}{\text{Total number of items}}. \tag{2}$$

The score ranges from 0 (no items satisfied) to 1 (all items satisfied). Scoring is performed automatically by the judge agent during execution and enters the composite refinement score of Equation 1.

It is worth noting that MAGMAScore is the metric that the orchestrator agent uses to refine future runs. It is therefore not used as an independent metric to evaluate workflow quality for either MAGMA or any of the tested baselines. We reserve that role to the rubric in Section 4.2, which is a separate instrument that is scored by human assessors to evaluate system performance.

3.3 MAGMA-SUITE: DATASETS & TASKS

We developed MAGMA-Suite consisting of 10 case studies using real-world datasets covering four modalities: tabular electronic health records data, chest X-ray images, clinical notes, and ECG. We selected a diverse set of modalities to ensure that the data and tasks have varying clinical relevance and levels of difficulty. The seven source datasets used in the compilation of the suite are publicly available, including (i) MIMIC-IV (Johnson et al., 2023b); (ii) MIMIC-CXR (Johnson et al., 2019); (iii) MIMIC-ECG (Johnson et al., 2023b); (iv) MIMIC-Note (Johnson et al., 2023a); (v) PhysioNet 2019 Challenge (Reyna et al., 2019); (vi) NIH ChestX-Ray14 (Wang et al., 2017); and (vii) PTB-XL (Wagner et al., 2020). In case studies 1 - 4, the tasks involve unimodal model training, while case studies 5 - 10 reflect all possible combinations of bimodal model training scenarios. As such, we believe the MAGMA-Suite offers sufficient coverage to assess the clinical capabilities of existing systems. The cohort, inclusion criteria, label derivation, prediction horizons, missing-data policy, and leakage controls are specified per case study in Appendix C.

Note that we apply no benchmark-level imputation or row dropping and propagate each dataset's inherent missingness to the evaluated systems, since data cleaning and missing-data handling are core competencies that pre-imputation would remove from measurement. Patient-level leakage is controlled at the split level by partitioning at the patient rather than the record level, which matters particularly for the six multimodal case studies where a record-level split could place one patient's modalities on both sides of the partition. Any occurrence of label leakage is recorded as a failed run, since avoiding it is part of what a competent autonomous system is expected to do.

4 EXPERIMENTAL SETUP

4.1 BASELINE MODELS

We evaluated the performance of eight baseline systems on the MAGMA-Suite. The objective is to quantify the predictive performance of existing systems, measure their workflow efficiency, assess their robustness, and establish a controlled comparison for future work. We evaluated three baseline configurations: (i) single-agent baselines: GPT-5.5, Claude-Opus-4.8, Gemini-3.1-Pro, DeepSeek-V4; (ii) general-purpose multi-agent baselines, allowing us to measure the effect of modularization and iterative critique without domain constraints (MLZero); (iii) and medical-centric baselines: CliMB-DC, HACHI, and HealthFlow.

We define a run as *completed* only if it produced a model artifact from which a reproducible prediction could be generated on the held-out partition. As a result, runs that failed to generate code, failed during training, produced no restorable model, crashed at inference, or hard-coded the exposed data path into their inference script are recorded as failures and counted as such. Given that this criterion is strict and several baselines fail frequently, we report both predictive performance conditional on completion and a completion-weighted score in which failed runs are assigned a value of zero, so that practical reliability is not hidden behind conditional averages. HealthFlow, MLZero, HACHI, and CliMB-DC were executed on case studies 1–4 only, because all four are unimodal systems: they accept a single input modality per task and cannot construct a pipeline for the six bimodal case studies 5-10, which require a model to consume two modalities jointly given their unimodal capabilities. Their completion rates are computed over the reduced grid.

4.2 EXPERT EVALUATION

We evaluate each system by human experts using a rubric derived from the functional requirements framework for clinical AutoML platforms (Scott et al., 2024): modality handling, data cleansing and transformation, feature engineering, model comparison, metric diversity, optimisation, scoring pipelines, and documentation. Each item is scored Compliant (2), Partially compliant (1), or Non-compliant or not applicable (0), giving a maximum of 16 points, which we report normalised to $[0, 1]$. The full rubric is summarized in Appendix D. The evidence packages given to the assessors, the candidate selection rule, and the blinding and randomisation procedure are described in Appendix E.

4.3 IMPLEMENTATION DETAILS

MAGMA was implemented in Python 3.11 using the Strands Agents SDK, with GPT-5.4 as the base LLM for all agents. Each run used a dedicated directory storing generated code, handoff files, processed data, models, predictions, metrics, judge outputs, and retry instructions; agents communicated through these artifacts rather than direct free-form messaging.

Each system–case-study–template configuration was executed in three independent runs. We evaluated four prompt templates varying requester framing (technical or clinical) and whether the evaluation rubric was supplied: Technical, Technical + Rubric, Clinical, and Clinical + Rubric. The exact templates are provided in Appendix H.1. The single-agent grid therefore comprised $4 \times 10 \times 4 \times 3 = 480$ runs, while MAGMA used the same ten case studies, four templates, and three runs. It is worth noting that...

We set $\lambda = 0.5$, $\eta = 1.0$, $p = 3$, $\epsilon = 0.01$, and allowed a maximum of 10 attempts per run. Appendix F examines sensitivity to λ and η. Baselines were run on one NVIDIA A100 GPU with 80 GB memory and 200 GB system memory; MAGMA used two NVIDIA Tesla V100 GPUs with 32 GB each.

5 RESULTS

5.1 EXECUTION RELIABILITY AND OVERALL PERFORMANCE

Figure 2 presents every system on the axes the benchmark measures. Detailed results are in Table S9. We report execution reliability, given that under a fixed held-out protocol, a significant fraction of

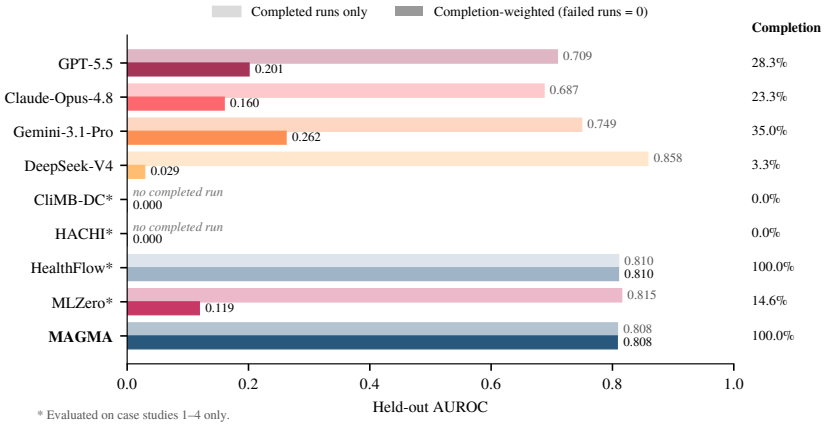


Figure 2: Held-out AUROC per system, averaged over completed runs only (light bars) and completion-weighted with failed runs scored as zero (solid bars). The gap between the two reflects the cost of failed runs; run completion is on the right. *Evaluated on case studies 1–4 only.

runs never reach the point of producing a reliable model. Across the 480 single-agent runs, 108 (22.5%) produced a reproducible held-out set predictions, and completion ranged from 35.0% for Gemini-3.1-Pro to 3.3% for DeepSeek-V4. Additionally, coverage is uneven across tasks: no single-agent baseline completed a run on case study 8, and only GPT-5.5 completed any run on case study 3, in one of 12 attempts. Per-system completion counts are given in Table S10 and per-case-study counts in Tables S11 and S12 (Appendix G), and Table S13 decomposes the failures. The dominant mode of failure is during training (218 of 552 attempted baseline runs), but the most instructive is a hard-coded exposed data path (60 runs, concentrated in Gemini-3.1-Pro and Claude-Opus-4.8), in which a system wrote the training-partition path directly into its inference script and could therefore not be evaluated on unseen patients at all. This is a failure that a benchmark permitting self-constructed splits cannot observe, since under such a protocol the system scores itself on data it has already seen.

The two columns of held-out AUROC in Table S9 show the importance of accounting for failures. Conditional on completing a run, DeepSeek-V4 records the highest predictive performance of any baseline (0.858, from four completed runs). However, once its 3.3% completion rate is accounted for, it records the lowest completion-weighted AUROC (0.029) of any baseline that completed at least one run.

MAGMA completed all 120 of its runs, with a mean held-out AUROC of 0.808 and AUPRC of 0.503 across all ten case studies. Because HealthFlow was evaluated on case studies 1-4 only, the two are not directly comparable in Table S9. Restricted to case studies 1-4, MAGMA's mean held-out AUROC is 0.834 compared to HealthFlow's AUROC of 0.810. Per case study, MAGMA records the highest completion-weighted AUROC in 9 of the 10 case studies, is the only system to complete any run on case study 8, and is outperformed only on case study 2, by HealthFlow.

5.2 Self-Reported versus Held-Out Performance

Where a system both completed a run and reported its own validation performance, we compared that self-reported validation AUROC against the held-out AUROC we computed independently (Table S14 and Figure S1). GPT-5.5 and Claude-Opus-4.8 over-report by $+0.053$ and $+0.081$ respectively, and their self-reports track the held-out result poorly ($r = 0.43$ and $r = 0.59$). Gemini-3.1-Pro reports a usable figure in only 19% of its completed runs, and DeepSeek-V4 and MLZero in none. By contrast, HealthFlow, which persists evaluation artifacts, is almost exactly calibrated to its own held-out result ($+0.002$, $r = 0.98$), followed by MAGMA ($+0.025$, $r = 0.75$, with a self-report in every completed run).

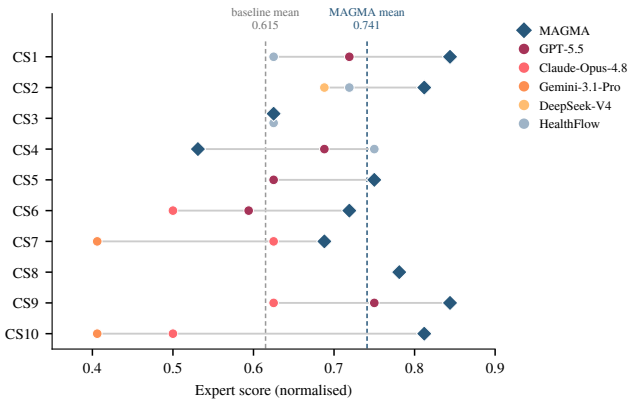


Figure 3: Blinded expert score per case study for MAGMA (diamonds) and the baselines scored on the same case study (circles). Dashed lines mark the MAGMA and baseline means; tied scores are offset vertically so that both remain visible.

5.3 Expert Adjudication Results

Figure 3 summarizes the blinded expert adjudication results, with detailed results summarized in Table S15. MAGMA received the highest expert score in 8 of the 10 case studies, tied with HealthFlow on case study 3, and was outscored on case study 4. Averaged across case studies MAGMA scores 0.741, against 0.615 across the sixteen baseline packages. Given these results, we note that no baseline produced an evaluable package for case study 8, so MAGMA's score there is uncontested rather than comparative. Additionally, on case studies 5 and 9 the strongest available baselines have a held-out AUROC of 0.5, so the comparison on those case studies is against models that do not discriminate. Furthermore, the case study 4 result, the only case study MAGMA loses, is also the site of the assessors' largest single disagreement (5 versus 12 of 16). Inter-rater agreement, the assessors' scoring conventions, and the item-level breakdown are reported in Appendix E.

5.4 Prompt Sensitivity

Completion varies by prompt template from 16.7% to 30.8% across the single-agent baselines (Table S17). MAGMA is far less sensitive on predictive performance, ranging from 0.794 to 0.818 held-out AUROC across the four templates, but its MAGMAScore responds in a consistent direction: 0.979 and 0.977 when the rubric is supplied in the prompt against 0.951 and 0.957 when it is not.

6 Discussion

Our results highlight three considerations for evaluating autonomous clinical machine-learning systems: execution reliability, independent evaluation, and the effect of guideline-grounded refinement.

Execution reliability is part of system performance. A central finding is the large gap between predictive performance conditional on successful execution and performance once failed runs are included. Across the single-agent baselines, only 22.5% of attempted runs produced reproducible held-out predictions. DeepSeek-V4 illustrates this distinction clearly: it achieved a conditional AUROC of 0.858, but its 3.3% completion rate reduced its completion-weighted AUROC to 0.029. Evaluating only successful runs can therefore substantially overstate the practical capability of autonomous ML systems. For end-to-end systems, failures during training, model restoration, or inference are part of the capability being evaluated rather than external implementation details. This motivates reporting execution reliability alongside conventional predictive metrics.

Independent evaluation and auditability are necessary. Self-reported validation performance also differed from independently computed held-out performance for several systems, while some systems failed to produce usable evaluation artifacts at all. Autonomous ML systems should therefore

be evaluated using models and scoring pipelines that can be reproduced independently on unseen data, rather than relying only on metrics reported by the system itself.

Guideline-grounded refinement affects workflow quality. MAGMAScore is used directly within MAGMA's refinement loop and therefore cannot serve as independent evidence that refinement improves workflow quality. The blinded expert adjudication provides a separate evaluation signal. MAGMA received the highest expert score in 8 of 10 case studies, with a mean score of 0.741 compared with 0.615 across the baseline packages. Importantly, the item-level results show that the differences are concentrated in model comparison, optimisation, metric diversity, and documentation, rather than basic modality handling or scoring-pipeline construction. This suggests that guideline-grounded refinement primarily improves the deliberative and reporting components of the workflow, rather than simply improving code generation or data handling.

Prompt sensitivity matters for benchmark design. Across the single-agent baselines, completion ranged from 16.7% to 30.8% across the four prompt templates, while MAGMA was more stable in predictive performance. Its MAGMAScore also increased when the evaluation rubric was explicitly provided. These results suggest that evaluation using a single prompt formulation may conflate prompt sensitivity with underlying system capability, although these effects are descriptive and their statistical significance has not been established.

Limitations. Our study has several limitations. Calibration and subgroup performance are not reported comparatively, and the expert adjudication includes only two assessors. MAGMA is evaluated in one primary configuration, with only limited sensitivity analysis, and all case studies use publicly available datasets. Future work should evaluate broader configurations, institutional data, subgroup robustness, privacy, security, and prospective clinical settings.

7 Conclusion

We introduced MAGMA, a multimodal agentic system for end-to-end clinical model development, together with MAGMA-Suite and a guideline-grounded refinement framework. MAGMA completed all benchmark runs and produced strong completion-weighted predictive performance and favourably rated workflows under blinded expert evaluation. More broadly, our results show that autonomous clinical ML systems should be evaluated on reliability, reproducibility, auditability, and workflow quality alongside predictive performance.

Ethics statement

This work uses only publicly available, de-identified datasets obtained under their respective data use agreements, and introduces no new human-subjects data collection. The expert adjudication was conducted by assessors not involved in this study.

AI use statement

In this work, we used generative AI tools as the subject of the study. All evaluated systems, including MAGMA, are built on large language models. Their use is documented in full in the relevant sections. We additionally used generative AI tools to assist with editing the text of this paper. All AI-assisted text was reviewed and revised by the authors. We take responsibility for the final content of this work, including all text, claims, and artifacts.

Reproducibility statement

The system architecture is described in detail, including implementation details, base model, seeds, and refinement hyperparameters. The benchmark protocol, including cohort definitions, label derivation, prediction horizons, and patient-level split methodology, is also provided. All datasets used are publicly available. Code for MAGMA, benchmark construction, and evaluation is available in our anonymized repository at `https://anonymous.4open.science/r/magma-B605/`.

REFERENCES

Federico Cabitza. Why accuracy isn't enough. rethinking model evaluation in clinical ai with a user-centered utility metric. In Cristian Tommasino, Cristiano Russo, and Michele Bernardini (eds.), *Artificial Intelligence for Biomedical Data*, pp. 255–264, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-17216-7.

Gary S Collins, Jérémie F Cohen, Constanza L Andaur Navarro, and et al. Tripod+ai statement: updated guidance for reporting clinical prediction models that use regression or machine learning methods. *BMJ*, 385:q902, 2024. doi: 10.1136/bmj-q902. URL https://bmj.com.

José Guilherme de Almeida, Christina Messiou, Sam J Withey, Celso Matos, Dow-Mu Koh, and Nickolas Papanikolaou. Medical machine learning operations: a framework to facilitate clinical ai development and deployment in radiology. *European Radiology*, 35(11):6828–6841, 2025.

Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. Autogluon-tabular: Robust and accurate automl for structured data, 2020. URL https://arxiv.org/abs/2003.06505.

Haoyang Fang, Boran Han, Nick Erickson, Xiyuan Zhang, Su Zhou, Anirudh Dagar, Jiani Zhang, Ali Caner Turkmen, Cuixiong Hu, Huzefa Rangwala, Ying Nian Wu, Bernie Wang, and George Karypis. Mlzero: A multi-agent system for end-to-end machine learning automation, 2025. URL https://arxiv.org/abs/2505.13941.

Jean Feng, Avni Kothari, Patrick Vossler, Andrew Bishara, Lucas Zier, Newton Addo, Aaron Kornblith, Yan Shuo Tan, and Chandan Singh. Human-ai co-design for clinical prediction models. *Arxiv Preprint*, 2026. URL https://arxiv.org/abs/2601.09072.

Jack Gallifant, Majid Afshar, Saleem Ameen, Yindalon Aphinyanaphongs, Shan Chen, Giovanni Cacciamani, Dina Demner-Fushman, Dmitriy Dligach, Roxana Daneshjou, Chrystinne Fernandes, Lasse Hyldig Hansen, Adam Landman, Lisa Lehmann, Liam G. McCoy, Timothy Miller, Amy Moreno, Nikolaj Munch, David Restrepo, Guergana Savova, Renato Umeton, Judy Wawira Gichoya, Gary S. Collins, Karel G. M. Moons, Leo A. Celi, and Danielle S. Bitterman. The tripod-llm statement: A targeted guideline for reporting large language models use. *medRxiv*, 2024. doi: 10.1101/2024.07.24.24310930. URL https://www.medrxiv.org/content/early/2024/07/25/2024.07.24.24310930.

Jifan Gao, Mahmudur Rahman, John Caskey, Madeline Oguss, Ann O'Rourke, Randy Brown, Anne Stey, Anoop Mayampurath, Matthew M. Churpek, Guanhua Chen, and Majid Afshar. Moma: A mixture-of-multimodal-agents architecture for enhancing clinical prediction modelling, 2025. URL https://arxiv.org/abs/2508.05492.

H2O.ai. H2o automl: Automatic machine learning, 2026. URL https://github.com/h2oai/h2o-3. R package version 3.46.0.x.

Steven A. Hicks, Inga Strümke, Vajira Thambawita, Malek Hammou, Michael A. Riegler, Pål Halvorsen, and Sravanthi Parasa. On evaluation metrics for medical applications of artificial intelligence. *medRxiv*, 2021. doi: 10.1101/2021.04.07.21254975. URL https://www.medrxiv.org/content/early/2021/04/23/2021.04.07.21254975.

Alistair Johnson, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV-Note: Deidentified free-text clinical notes, 2023a. URL https://physionet.org/content/mimic-iv-note/2.2/.

Alistair EW Johnson, Tom J Pollard, Seth J Berkowitz, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Roger G Mark, and Steven Horng. Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific data*, 6(1):317, 2019.

Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1, 2023b.

Ilia Kopanichuk, Petr Anokhin, Vladimir Shaposhnikov, Vladimir Makharev, Ekaterina Tsapieva, Iaroslav Bespalov, Dmitry V. Dylov, and Ivan Oseledets. How to evaluate medical ai, 2025. URL https://arxiv.org/abs/2509.11941.

Xiangbin Meng, Xiangyu Yan, Kuo Zhang, Da Liu, Xiaojuan Cui Yaodong Yang, Muhan Zhang, Chunxia Cao, Jingjia Wang, Xuliang Wang, Yuan-Geng-Shuo Wang Jun Gao, Jia ming Ji, Zifeng Qiu, Muzi Li, Cheng Qian, Tianze Guo, Shuangquan Ma, Zeying Wang, Zexuan Guo, Youlan Lei, Chunli Shao, Wenyao Wang, Haojun Fan, , and Yi-Da Tang. The application of large language models in medicine: A scoping review. *Iscience*, 2024. URL https://www.sciencedirect.com/science/article/pii/S2589004224009350?via%3Dihub.

Brenda Y. Miao, Irene Y. Chen, Christopher YK Williams, Jaysón Davidson, Augusto Garcia-Agundez, Shenghuan Sun, Travis Zack, Suchi Saria, Rima Arnaout, Giorgio Quer, Hossein J. Sadaei, Ali Torkamani, Brett Beaulieu-Jones, Bin Yu, Milena Gianfrancesco, Atul J. Butte, Beau Norgeot, and Madhumita Sushil. The minimum information about clinical artificial intelligence checklist for generative modeling research (mi-claim-gen), 2024. URL https://arxiv.org/abs/2403.02558.

Matthew Reyna, Chris Josef, Russell Jeter, Supreeth Shashikumar, Benjamin Moody, M. Brandon Westover, Ashish Sharma, Shamim Nemati, and Gari D. Clifford. Early Prediction of Sepsis from Clinical Data: The PhysioNet/Computing in Cardiology Challenge 2019. *PhysioNet*, August 2019. doi: 10.13026/v64v-d857. URL https://doi.org/10.13026/v64v-d857. Version 1.0.0.

Evgeny Saveliev, Jiashuo Liu, Nabeel Seedat, Anders Boyd, and Mihaela van der Schaar. Towards human-guided, data-centric llm co-pilots, 2025. URL https://arxiv.org/abs/2501.10321.

Ian A Scott, Keshia R De Guzman, Nazanin Falconer, Stephen Canaris, Oscar Bonilla, Steven M McPhail, Sven Marxen, Aaron Van Garderen, Ahmad Abdel-Hafez, and Michael Barras. Evaluating automated machine learning platforms for use in healthcare, 2024. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC11165368/.

Farah Shamout, Tingting Zhu, and David A Clifton. Machine learning for clinical outcome prediction. *IEEE reviews in Biomedical Engineering*, 14:116–126, 2020.

Arun James Thirunavukarasu, Kabilan Elangovan, Laura Gutierrez, Refaat Hassan, Yong Li, Ting Fang Tan, Haoran Cheng, Zhen Ling Teo, Gilbert Lim, and Daniel Shu Wei Ting. Clinical performance of automated machine learning: a systematic review. *medRxiv*, pp. 2023–10, 2023.

Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I Lunze, Wojciech Samek, and Tobias Schaeffter. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific data*, 7(1):154, 2020.

Chi Wang, Qingyun Wu, Markus Weimer, and Erkang Zhu. Flaml: A fast and lightweight automl library, 2021. URL https://arxiv.org/abs/1911.04706.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017.

Yinghao Zhu, Yifan Qi, Zixiang Wang, Lei Gu, Dehao Sui, Haoran Hu, Xichen Zhang, Ziyi He, Junjun He, Liantao Ma, and Lequan Yu. Healthflow: A self-evolving ai agent with meta planning for autonomous healthcare research, 2025. URL https://arxiv.org/abs/2508.02621.

A Formal Specification of the Refinement Process

We formulate MAGMA as a finite-horizon, orchestrator-guided iterative refinement process. Each attempt $t \in \{0, \ldots, H-1\}$ begins with an execution context X_t containing the task specification, the artifacts produced by prior attempts, and any retry instructions. The orchestrator starts the first attempt from the task context and guides any subsequent attempts using the judge's verdict and the score from the preceding attempt:

$$A_t = \begin{cases} O(X_0), & t = 0, \\ O(X_t,\, F_{t-1},\, S_{t-1}), & t > 0, \end{cases}$$

where A_t denotes the execution or retry instructions routed by the orchestrator, F_{t-1} the judge's structured feedback, and S_{t-1} the composite refinement score. For each worker $w \in \{D, M\}$, denoting the data and model workers, the worker generates artifacts according to

$$W^w_{t,k+1} = A_w\big(X_t,\, A_t,\, q^w_{t,k}\big)\,, \qquad q^w_{t,k+1} = V_w\big(W^w_{t,k+1}\big)\,,$$

where V_w is the corresponding validator and k indexes validator-driven repairs within attempt t. If $q^w_{t,k+1}$ requests a repair, the orchestrator routes the feedback back to the respective worker. Once accepted, the data and model artifacts are denoted $\bar{D}_t$ and $\bar{M}_t$ and are returned to the orchestrator for evaluation by the judge, which produces structured feedback $F_t = J(\bar{D}_t, \bar{M}_t)$ together with the composite refinement score defined in Equation 1. If refinement continues, the orchestrator folds both into the next execution context, $X_{t+1} = U(X_t, A_t, F_t, S_t)$, where U corresponds in our implementation to writing new artifacts to the run directory. The process terminates at

$$\tau = \min\left\{ t \;:\; S_t \geq \eta \;\;\vee\;\; t = H-1 \;\;\vee\;\; \max_{j=t-p+1,\ldots,t} (S_j - S_{t-p}) < \epsilon \right\},$$

where η is the target composite refinement score, H the maximum number of attempts, p the plateau window, and ϵ the minimum improvement required over that window. We note explicitly that no model parameters are updated and nothing is carried from one attempt sequence to the next: refinement operates entirely through the execution context, and we therefore describe the mechanism as context-based iterative refinement rather than reinforcement learning.

B MAGMAScore Items

MAGMAScore is computed from the 22 items in Table S1. Each is scored Yes or No by the judge agent against the run's artifacts, and MAGMAScore is the proportion satisfied. Source codes: **T+AI** = TRIPOD+AI item; **T-LLM** = TRIPOD-LLM item; **MIC** = MI-CLAIM-GEN section; **CA** = clinical AutoML functional requirement (Scott et al., 2024).

C Dataset Construction and Held-out Test Set Partitioning

In this section, we demonstrate the methodology used to assemble our MAGMA-Suite case studies from the seven publicly available clinical dataset, and how a patient-disjoint held-out partition was derived. All randomized operations use a fixed seed ($\text{seed} = 42$) and are deterministic and reproducible.

This appendix specifies, for each case study, the cohort definition, the label derivation rule, and the prediction time horizon. The cohort is defined identically across all ten case studies: all patients with data in every predictor modality the task requires and with a definable label. Inclusion is deliberately broad, restricted only to presence in the relevant modality, so that predictor-side data handling remains a capability under test. Labels are anchored to an established clinical or reporting standard rather than constructed anew. We apply no benchmark-level imputation or row dropping, propagating each dataset's inherent missingness to the evaluated systems. Leakage is controlled at the split level by patient-level partitioning, described in Appendix C.

We note as a limitation that these definitions are anchored at the source level, and that we have not independently audited edge cases within our own cohort construction beyond that source-level validation.

Table S1: The 22 MAGMAScore items used by the judge agent.

#	Item	Source
1. Problem framing		
1.1	Does the system understand and explain the healthcare context and rationale for developing the prediction model?	T+AI 3a–c
2. Data preparation		
2.1	Does the system describe data pre-processing and quality checking steps?	T+AI 7
2.2	Is data pre-processing applied consistently across relevant sociodemographic groups?	T+AI 7, T-LLM 5d
2.3	Does the system apply data cleansing and/or transformation (e.g. normalisation, encoding, imputation strategy)?	T+AI 7, 11; CA FR03
2.4	Does the system explicitly recognise and address data leakage in the dataset (when applicable)?	T+AI 12a, MIC Part 2
3a. Data partitioning and feature engineering		
3a.1	Does the system describe how data were partitioned into training, validation, and test sets, where the split is not provided?	T+AI 12a
3a.2	Is the train/validation/test split performed at the patient level rather than the row level, to prevent leakage (when applicable)?	T+AI 12a, MIC Part 2
3a.3	Does the system perform feature engineering on the input data (when applicable)?	T+AI 12b, CA FR02
3b. Model building and selection		
3b.1	Does the system specify the model type(s) used?	T+AI 12c
3b.2	Does the system compare multiple model types to determine the best fit for the task?	T+AI 12c, CA FR06
3b.3	Does the system use or consider ensemble methods or blended models (when applicable)?	T+AI 12c, CA FR05
3b.4	Does the system apply supervised learning appropriately for the prediction task?	T+AI 12c, CA FR04
3b.5	Does the system perform hyperparameter tuning?	T+AI 12c, CA FR08
3c. Heterogeneity across settings and class imbalance		
3c.1	If applicable, is heterogeneity in model performance across subgroups or settings handled and quantified?	T+AI 12d
3c.2	Does the system explicitly recognise and address class imbalance in the dataset?	T+AI 13
4. Model specification		
4.1	Does the system provide the full model specification (architecture, hyperparameters, code)?	T+AI 22
4.2	Does the system specify the model output type (probability, risk score, binary label)?	T+AI 15
5. Model performance and evaluation		
5.1	Can the system handle multiple data modalities appropriate to the use case?	CA FR01
5.2	Does the system report model performance estimates with appropriate evaluation metrics?	T+AI 23a
5.3	Does the system justify the evaluation approach and primary metrics used for the task?	T+AI 23a, CA FR13
5.4	Does the system produce visualisations to support performance interpretation or model explainability?	MIC Parts 3–4
5.5	Does the system report compute requirements or cost proxies?	T-LLM 12

C.1 Source Datasets

As mentioned in Table S2, the MAGMA-Suite benchmark draws on seven source datasets that were used to construct ten case studies spanning tabular, waveform, imaging, and clinical-text modalities. MIMIC-Note is treated as two datasets (discharge summaries and radiology reports) because they have distinct record populations, but they share a single subject-level split assignment for cross-modality consistency. Table S5 summarizes the inventory.

Table S2: Overview of clinical tasks with their corresponding modalities and datasets.

Case Study	Task	Modality	Dataset
1	Sepsis onset	EHR	PhysioNet 2019
2	COPD diagnosis	Text	MIMIC-Note
3	Pneumonia	Imaging	ChestX-ray14
4	Arrhythmia	Signal	PTB-XL
5	Pleural effusion	CXR + EHR	MIMIC-CXR + IV
6	In-hospital mortality	EHR + ECG	MIMIC-IV + ECG
7	30-day readmission	EHR + Text	MIMIC-IV + MIMIC-Note
8	Cardiomegaly	CXR + ECG	MIMIC-CXR + ECG
9	30-day readmission	CXR + Text	MIMIC-CXR + MIMIC-Note
10	30-day readmission	ECG + Text	MIMIC-ECG + MIMIC-Note

Table S3: Prediction time horizon per case study.

Case study	Time horizon
CS1	Fixed: 6-hour sepsis onset window
CS2, CS3, CS4, CS5, CS6, CS8	Encounter-bounded (outcome defined over the clinical encounter)
CS7, CS9, CS10	Fixed: 30-day readmission window

For MIMIC-IV we use release v1.0 as the authoritative source and split on the raw core/patients.csv subject list. For datasets shipping their own canonical splits such as (ChestX ray14's train_list.txt/ test_list.txt, MIMIC-CXR's *-split.csv files, PTB XL's strat_fold column), those native splits are ignored in favor of the unified patient-level partition described below, so that all systems are evaluated on identical, contamination-controlled held-out data.

C.2 Split Methodology

All splits are performed at the patient level and never the record. This guarantees that no patient contributes data to more than one partition, eliminating patient-level leakage for systems that consume multiple records or modalities per patient. The five MIMIC-derived datasets (MIMIC-IV, MIMIC-CXR, MIMIC-ECG, and the two MIMIC-Note variants) share the `subject_id` namespace. We form the union of all subject identifiers appearing in any of the five sources (386296 unique subjects) and assign each subject a single split label. This way, since every MIMIC dataset consults the same subject $\rightarrow$ split table, a subject is mechanically prevented from landing in different partitions across modalities. The three remaining datasets (PhysioNet 2019, ChestX-ray14, PTB XL) have disjoint patient namespaces and are split independently. Our splitting procedure for each split pool is as follows: (i) collect the unique patient identifiers; (ii) sort them (making the subsequent shuffle reproducible irrespective of set-iteration order); (iii) apply a single deterministic permutation under $\text{seed} = 42$; (iv) slice the permuted list into $85\%/15\%$ for train/test, with any rounding remainder absorbed by the test partition so the two slices sum exactly to the pool size. As such, our benchmark is released as a two-way **exposed / held-out** partition, with the *exposed* partition being the train subjects, and the *held-out* partition being the test subjects.

C.3 Per-Dataset Partition Statistics

Table S6 reports the patient-level partition for every dataset.

In our splitting procedure, for every dataset we verified that (a) each source's full patient set is covered by an assignment with zero unassigned ("missing") subjects; (b)the per-source train/test proportions remain within sampling variance of the target $85/15$, and that the uniform random split induces no skew; and (c) the two partitions are mutually disjoint at the patient level. For the independent datasets we additionally confirmed that label prevalence is preserved across partitions. Representative results: ChestX-ray14 "No Finding" prevalence was $0.538/0.538$ across train/test and pneumonia prevalence $0.0129/0.0117$; PTB-XL normal-rhythm prevalence was $0.438/0.430$; for PhysioNet 2019 both source hospitals (Set A and Set B) contributed proportionally to both partitions. Subject-overlap analysis showed that fewer than 2% of subjects in MIMIC-CXR, MIMIC-ECG, or MIMIC-Note are absent from the MIMIC-IV v1.0 patient table (maximum 1.81%, MIMIC-ECG), consistent with those datasets being released against a later MIMIC-IV minor version; these subjects are retained and assigned splits.

Table S4: Label derivation per case study.

CS	Label definition
CS1	`SepsisLabel==1` at any hour (native PhysioNet 2019 Challenge label, shifted 6 hours upstream)
CS2	Broad COPD ICD definition (ICD-9: 491/492/496; ICD-10: J40–J44); any admission positive → patient positive
CS3	Literal token "Pneumonia" in the Finding Labels field (word-boundary match); any image positive → patient positive
CS4	PTB-XL SCP codes mapped through `diagnostic_class` into five super-classes, with a per-label OR across a patient's ECGs
CS5, CS8	CheXpert labeler output under a U-Zeros policy for uncertain findings; any study positive → patient positive
CS6	`hospital_expire_flag`; any admission positive → patient positive
CS7, CS9, CS10	Next admit-time within 30 days of discharge-time with the index admission non-death; any eligible admission positive → patient positive

Table S5: Source dataset inventory. MIMIC-Note contributes two record sets, discharge summaries and radiology reports, which are listed separately here because they have distinct record populations; they share a single subject-level split assignment, so the seven source datasets yield eight rows. "Records" is the native row/file count; "Patients" is the number of unique patient identifiers used as the unit of splitting.

Dataset	Patient ID	Records	Patients
MIMIC-IV (v1.0)	`subject_id`	382278	382278
MIMIC-CXR	`subject_id`	377110	65379
MIMIC-ECG	`subject_id`	800035	161352
MIMIC-Note (discharge)	`subject_id`	331793	145914
MIMIC-Note (radiology)	`subject_id`	2321355	237427
PhysioNet 2019 (Sepsis)	filename `p`⟨`id`⟩	40336	40336
ChestX-ray14 (NIH)	`Patient ID`	112120	30805
PTB-XL	`patient_id`	21799	18869

C.4 Reproducibility

Split assignments are stored as flat lookup tables (⟨`dataset`⟩`_*.csv`) mapping each patient identifier to its partition label. The two way ($85/15$, seed 42) assignments are in the released tables, so that the derivation in §C.2 can be reproduced exactly. We provide the code for generating the split assignments in our github page.

D Evaluation Rubric

The rubric introduced in Section 4.2 is the external evaluation instrument applied uniformly to every system in the benchmark, including the baselines and MAGMA, and scored identically by the two human assessors. It is adapted from Table 3 of the functional-requirements framework for clinical AutoML platforms (Scott et al., 2024), from which we selected the eight requirements that can be assessed from the artifacts an autonomous system produces for a single modelling task.

It is a separate instrument from MAGMAScore (Appendix B): MAGMAScore comprises 22 binary items scored automatically inside the refinement loop, whereas the rubric below comprises eight items scored on a three-point scale by a reader of the evidence package, and no MAGMAScore item appears in it. The two nonetheless draw on overlapping bodies of clinical reporting guidance, which we regard as unavoidable for any clinically meaningful rubric.

Table S6: Patient-level partition counts. "Exposed" = Train; "Held-out" = Test. All MIMIC datasets share one `subject_id` assignment derived over the 386,296-subject union.

	Released	
Dataset	Exposed	Held-out
MIMIC-IV	324,946	57,332
MIMIC-CXR	55,639	9,740
MIMIC-ECG	137,053	24,299
MIMIC-Note (discharge)	123,986	21,928
MIMIC-Note (radiology)	201,664	35,763
PhysioNet 2019	34,285	6,051
ChestX-ray14	26,185	4,620
PTB-XL	16,038	2,831

Table S7: The eight-item evaluation rubric, scored Compliant (2), Partially compliant (1), or Non-compliant / not applicable (0), for a maximum of 16 points per run. Adapted from Scott et al. (Scott et al., 2024), Table 3.

#	Item	What it assesses
1	Modality handling	Supports the handling of the input data modality required by the task.
2	Data cleansing and transformation	Demonstrates data cleansing and data transformation capability.
3	Feature engineering	Performs feature engineering on the imported datasets.
4	Model comparison	Applies multiple models to the given dataset to determine the best fit for purpose.
5	Metric diversity	Applies different metrics when developing and evaluating the model.
6	Optimisation	Provides for additional tuning or optimisation of the developed model.
7	Scoring pipelines	Produces a scoring pipeline capable of making predictions on newly acquired data.
8	Documentation	Generates documentation detailing the model and its findings.

E Expert Adjudication: Protocol and Detailed Results

Evidence packages. Each run is represented by a prespecified evidence package containing the generated model code, the run logs, and all accompanying artifacts, assembled identically for every system.

Blinded expert adjudication. For each of the ten case studies we selected MAGMA together with the two baselines achieving the highest held-out AUROC on that case study, so that MAGMA is compared against the technically strongest available comparators. Where fewer than two baselines produced a completed run, fewer comparators were available: CS3 and CS5 have one and CS8 has none, and on CS5 and CS9 the strongest available baselines are themselves degenerate at a held-out AUROC of exactly 0.500. Packages were anonymised by stripping system names, file paths, MAGMAScore values, and all LLM judge output, and candidate order was randomised independently for each assessor.

The two assessors have significant clinical machine-learning expertise, and neither was involved in the development of MAGMA or in running any system. They scored all 26 packages independently on the eight-item rubric.

The two assessors showed moderate inter-rater agreement (quadratic-weighted Cohen's $\kappa = 0.42$; 55.1% exact and 94.2% within-one-point agreement across 207 paired item scores on all 26 packages). Their disagreements were almost exclusively adjacent, indicating consistent ordinal judgement despite

differing strictness, with one assessor marginally more lenient overall (mean item score 1.40 versus 1.27).

The item-level breakdown (Appendix G, Table S18) localises MAGMA's advantage rather than leaving it aggregate. MAGMA outscores the baselines on model comparison (0.90 versus 0.22), optimisation (1.32 versus 0.53), documentation (1.70 versus 1.03), and metric diversity (1.55 versus 1.28), and is marginally weaker on modality handling (1.75 versus 1.81), data cleansing (1.50 versus 1.59), and scoring pipelines (1.70 versus 1.94). This shows that the advantage of our system lies in the parts of the workflow that reporting guidelines specifically govern, such as whether alternatives were compared, whether the search over them was deliberate, and whether the result was documented, rather than in raw data-handling competence, where contemporary coding systems are already strong.

F Sensitivity to the Refinement-Score Weighting λ

We initially set $\lambda = 0.5$ as a symmetric weighting, since AUROC and MAGMAScore are both bounded in $[0, 1]$. With the stopping threshold set to the maximum attainable score, $\eta = 1$, the terminal condition is invariant to λ for any $\lambda \in (0, 1)$, because $S_t = 1$ requires both components to equal 1 regardless of their weights. λ nonetheless affects the numerical score, the ranking of candidate attempts, and the information carried into subsequent retries. To examine this we repeated two case studies of contrasting difficulty at a relaxed threshold $\eta = 0.9$, sweeping $\lambda \in \{0, 0.2, 0.5, 0.7, 1\}$.

Table S8: Sensitivity to λ at a relaxed refinement threshold $\eta = 0.9$. "Best S_t" is the highest composite score attained across attempts; where it falls below η the run exhausted its attempt budget without terminating on the threshold.

λ	Attempts	Best attempt	Best MAGMAScore	Best S_t	Held-out AUROC
Case Study 2 (COPD diagnosis; easier)					
0.0	6	6	0.909	0.909	0.953
0.2	3	3	0.909	0.920	0.820
0.5	1	1	0.864	0.910	0.954
0.7	1	1	0.864	0.930	0.956
1.0	1	1	0.818	0.945	0.937
Case Study 9 (30-day readmission, CXR + text; harder)					
0.0	1	1	0.909	0.909	0.573
0.2	1	1	0.955	0.926	0.809
0.5	10	3	1.000	0.843	0.681
0.7	10	2	1.000	0.811	0.710
1.0	10	7	1.000	0.706	0.674

The two regimes behave in opposite directions, and the interaction with task difficulty is the substantive finding. On the harder Case Study 9, MAGMA attains a held-out AUROC of roughly 0.68–0.71; at $\lambda \geq 0.5$ the AUROC term caps S_t below 0.9 even at a perfect MAGMAScore, so the run exhausts its attempt budget. On the easier Case Study 2, $\lambda \geq 0.5$ reaches the threshold after a single attempt, which terminates the run before MAGMAScore can be improved further, while the lowest weightings continue refining and reach a higher MAGMAScore at the cost of additional attempts. λ and η must therefore be set jointly and with reference to the performance attainable on the task. We report $\lambda = 0.5$ as a reasonable neutral default rather than a universally optimal choice, and note that the $\eta = 1$ setting used in our main experiments removes the interaction entirely by requiring both components to saturate. This analysis covers two of the ten case studies.

G Additional Results

This appendix reports the per-case-study results underlying Section 5. Tables S11 and S12 give held-out AUROC and AUPRC over completed runs, with completion counts per cell; Table S16 gives the completion-weighted equivalents, in which failed runs are scored zero. Table S10 reports run completion per system and Table S13 the terminal outcome of every attempted run. Table S18 reports the item-level breakdown of the expert adjudication. Table S17 reports the effect of prompt

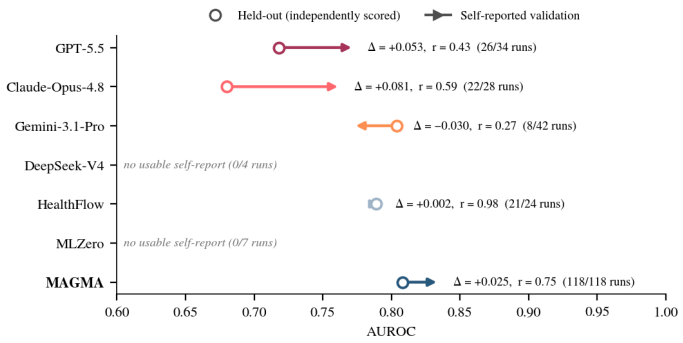


Figure S1: Self-reported validation AUROC against independently scored held-out AUROC. Each arrow runs from the held-out value (circle) to the value the system reported for itself, so rightward arrows denote over-reporting. Systems that produced no usable self-report are labelled.

Table S9: System-level summary across the MAGMA-Suite under the fixed held-out protocol. *Completion* is the proportion of attempted runs yielding a reproducible held-out prediction. *Conditional* AUROC is averaged over completed runs only; *completion-weighted* AUROC assigns zero to failed runs. Values are mean ± 1 standard deviation across runs. MAGMAScore and the expert score are averaged over the runs scored on each instrument, with the number of case studies in parentheses; the MAGMAScore column is computed on the clinical, no-criteria template for every system, since the other templates have not been scored for all baselines. *Evaluated on case studies 1–4 only. †Attempted on case studies 1–4 only, with no completed run.

		Held-out AUROC				
System	**Completion**	**AUROC (cond.)**	**AUROC-weighted**	**AUPRC (cond.)**	**MAGMAScore**	**Expert score**
GPT-5.5	28.3%	0.709 ±0.131	0.201 ±0.328	0.380 ±0.203	0.764 (5)	0.675 (5)
Claude-Opus-4.8	23.3%	0.687 ±0.151	0.160 ±0.300	0.420 ±0.162	0.682 (4)	0.562 (4)
Gemini-3.1-Pro	35.0%	0.749 ±0.123	0.262 ±0.366	0.390 ±0.264	0.636 (2)	0.406 (2)
DeepSeek-V4	3.3%	0.858 ±0.134	0.029 ±0.156	0.578 ±0.300	0.545 (1)	0.688 (1)
CliMB-DC†	0.0%	–	0.000	–	–	–
HACHI†	0.0%	–	0.000	–	–	–
HealthFlow*	97.9%	0.819 ±0.105	0.802 ±0.157	0.390 ±0.339	0.625 (4)	0.680 (4)
MLZero*	14.6%	0.815 ±0.068	0.119 ±0.292	0.330 ±0.326	0.383 (4)	–
MAGMA	100.0%	0.808 ±0.104	**0.808** ±0.104	0.503 ±0.299	**0.957 (10)**	**0.741** (10)

Table S10: Run completion under the fixed held-out protocol. A run counts as completed only if it produced a reproducible prediction on the held-out partition. *HealthFlow, MLZero, CliMB-DC, and HACHI were only executed on case studies 1 - 4 only as they are single modality baselines. †CliMB-DC and HACHI produced no completed run. Rates over the reduced grid are not directly comparable to the full grid.

System	**Attempted**	**Trained**	**Completed**	**Failed**	**Completion**
GPT-5.5	120	75	34	86	28.3%
Claude-Opus-4.8	120	78	28	92	23.3%
Gemini-3.1-Pro	120	86	42	78	35.0%
DeepSeek-V4	120	12	4	116	3.3%
CliMB-DC†	48	0	0	48	0.0%
HACHI†	48	0	0	48	0.0%
HealthFlow*	48	47	47	1	97.9%
MLZero*	48	15	7	41	14.6%
MAGMA	120	120	120	0	100.0%

template. All predictive values are computed on the fixed patient-disjoint held-out partition described in Appendix C, which no system observed during development.

Table S11: Held-out AUROC / AUPRC on the fixed patient-disjoint test partition, averaged over completed runs only, with (completed/attempted) in parentheses. Case studies 1–5. "–" denotes no completed run for that configuration.

System	CS1	CS2	CS3	CS4	CS5
GPT-5.5	0.834 / 0.250 (8/12)	0.957 / 0.836 (2/12)	0.510 / 0.032 (1/12)	0.692 / 0.466 (6/12)	0.543 / 0.296 (3/12)
Claude-Opus-4.8	0.814 / 0.415 (5/12)	0.923 / 0.774 (2/12)	– (0/12)	0.500 / 0.316 (1/12)	0.500 / 0.274 (1/12)
Gemini-3.1-Pro	0.790 / 0.084 (10/12)	0.812 / 0.558 (9/12)	– (0/12)	0.594 / 0.414 (6/12)	– (0/12)
DeepSeek-V4	0.781 / 0.358 (2/12)	0.935 / 0.798 (2/12)	– (0/12)	– (0/12)	– (0/12)
CliMB-DC	– (0/12)	– (0/12)	– (0/12)	– (0/12)	–
HACHI	– (0/12)	– (0/12)	– (0/12)	– (0/12)	–
HealthFlow	0.817 / 0.143 (12/12)	0.924 / 0.740 (12/12)	0.694 / 0.029 (12/12)	0.845 / 0.673 (11/12)	–
MLZero	0.760 / 0.069 (4/12)	0.888 / 0.678 (3/12)	– (0/12)	– (0/12)	–
MAGMA	0.827 $\pm$0.072 / 0.247 $\pm$0.207	0.897 $\pm$0.099 / 0.655 $\pm$0.208	0.697 $\pm$0.033 / 0.026 $\pm$0.004	0.920 $\pm$0.058 / 0.856 $\pm$0.068	0.834 $\pm$0.034 / 0.639 $\pm$0.074

Table S12: Held-out AUROC / AUPRC, case studies 6–10. Conventions follow Table S11. No single-agent baseline produced a completed run on CS8. HealthFlow, MLZero, CliMB-DC and HACHI were executed on case studies 1-4 only.

System	CS6	CS7	CS8	CS9	CS10
GPT-5.5	0.715 / 0.156 (3/12)	0.715 / 0.526 (5/12)	– (0/12)	0.500 / 0.391 (2/12)	0.625 / 0.415 (4/12)
Claude-Opus-4.8	0.764 / 0.200 (5/12)	0.779 / 0.599 (4/12)	– (0/12)	0.525 / 0.414 (9/12)	0.642 / 0.432 (1/12)
Gemini-3.1-Pro	0.741 / 0.168 (4/12)	0.794 / 0.627 (9/12)	– (0/12)	– (0/12)	0.644 / 0.431 (4/12)
DeepSeek-V4	– (0/12)	– (0/12)	– (0/12)	– (0/12)	– (0/12)
CliMB-DC	–	–	–	–	–
HACHI	–	–	–	–	–
HealthFlow	–	–	–	–	–
MLZero	–	–	–	–	–
MAGMA	0.863 $\pm$0.131 / 0.407 $\pm$0.346	0.786 $\pm$0.069 / 0.436 $\pm$0.077	0.840 $\pm$0.038 / 0.899 $\pm$0.034	0.676 $\pm$0.039 / 0.411 $\pm$0.053	0.723 $\pm$0.063 / 0.425 $\pm$0.081

Table S13: Terminal outcome of every attempted baseline run, over the full grid of four prompt templates and three runs per configuration ($n = 576$: 120 per single-agent system, 48 for HealthFlow and MLZero). `hardcoded_exposed_path` denotes runs that wrote the exposed training path into their inference script, making held-out inference impossible. CliMB-DC and HACHI are omitted because no run reached a terminal outcome that could be classified; both are counted as complete failures in Table S10, which is why the attempted totals there exceed 576.

Outcome	GPT-5.5	Claude	Gemini	DeepSeek	HealthFlow	MLZero	Total
completed (scored)	34	28	42	4	47	7	139
training failed	45	42	34	97	1	0	218
prediction crash	29	21	8	4	0	0	62
hardcoded exposed path	1	25	34	0	0	0	60
no model produced	0	0	0	0	0	21	21
generation failed	0	0	0	13	0	0	13
prediction script missing	10	1	1	0	0	0	12
no model artifact	0	0	0	0	0	6	6
no restorable model	0	0	0	0	0	6	6
leakage contaminated	0	0	0	0	0	6	6
label column mismatch	0	2	1	0	0	0	3
timeout	1	1	0	1	0	0	3
load error	0	0	0	0	0	1	1
model missing	0	0	0	1	0	0	1
model not fitted	0	0	0	0	0	1	1
Total	120	120	120	120	48	48	552

Model comparison is the weakest item across the benchmark as a whole (mean 0.48 of a possible 2.0), and optimisation is similarly weak (0.82); scoring pipelines (1.85) and modality handling (1.79) are near-ceiling for every system. Contemporary agentic systems are therefore competent at the mechanics of building and scoring a model, and weak at the deliberation and documentation that clinical reporting standards require. MAGMA narrows the second gap without closing it.

Table S14: Self-reported validation AUROC versus independently scored held-out AUROC, over completed runs. Δ is self-reported minus held-out, so positive values denote over-reporting.

System	Runs with self-report	Self-reported (validation)	Held-out	Δ	Pearson r
GPT-5.5	26/34 (76%)	0.771	0.718	$+0.053$	0.43
Claude-Opus-4.8	22/28 (79%)	0.761	0.680	$+0.081$	0.59
Gemini-3.1-Pro	8/42 (19%)	0.774	0.804	-0.030	0.27
DeepSeek-V4	0/4	–	–	–	–
HealthFlow	42/48 (88%)	0.791	0.789	$+0.002$	0.98
MLZero	0/7	–	–	–	–
MAGMA	120/120 (100%)	0.833	0.808	$+0.025$	0.75

Table S15: Blinded expert adjudication. Each package was scored independently by two assessors on eight items (Compliant $= 2$, Partially compliant $= 1$, Non-compliant or not applicable $= 0$; maximum 16). "Expert score" is the mean assessor total normalised to $[0, 1]$. † one item could not be scored by one assessor. System identity is shown for reporting only; it was hidden during adjudication, and candidate order was randomised independently per assessor.

CS	System	Assessor A	Assessor B	Expert score
CS1	**MAGMA**	16	11	**0.844**
CS1	GPT-5.5	12	11	0.719
CS1	HealthFlow	10	10	0.625
CS2	**MAGMA**	14	12	**0.812**
CS2	HealthFlow	10	13	0.719
CS2	DeepSeek-V4	10	12	0.688
CS3	**MAGMA**†	11	9	0.625
CS3	HealthFlow	10	10	0.625
CS4	HealthFlow	11	13	**0.750**
CS4	GPT-5.5	10	12	0.688
CS4	**MAGMA**	5	12	0.531
CS5	**MAGMA**	13	11	**0.750**
CS5	GPT-5.5	9	11	0.625
CS6	**MAGMA**	10	13	**0.719**
CS6	GPT-5.5	7	12	0.594
CS6	Claude-Opus-4.8	7	9	0.500
CS7	**MAGMA**	10	12	**0.688**
CS7	Claude-Opus-4.8	7	13	0.625
CS7	Gemini-3.1-Pro	5	8	0.406
CS8	**MAGMA**	13	12	**0.781**
CS9	**MAGMA**	13	14	**0.844**
CS9	GPT-5.5	12	12	0.750
CS9	Claude-Opus-4.8	10	10	0.625
CS10	**MAGMA**	13	13	**0.812**
CS10	Claude-Opus-4.8	8	8	0.500
CS10	Gemini-3.1-Pro	6	7	0.406
MAGMA mean across case studies				**0.741**
Baseline mean across 16 packages				0.615

Table S16: Completion-weighted held-out AUROC / AUPRC, in which every failed run is assigned a value of zero and the mean is taken over all attempted runs for that configuration. This separates model quality conditional on execution from practical reliability. *Executed on case studies 1–4 only. †Attempted on case studies 1–4 only, with no completed run.

System	CS1	CS2	CS3	CS4	CS5	CS6	CS7	CS8	CS9	CS10
GPT-5.5	0.556/0.167	0.159/0.139	0.042/0.003	0.346/0.233	0.136/0.074	0.179/0.039	0.298/0.219	0.000/0.000	0.083/0.065	0.208/0.138
Claude-Opus-4.8	0.339/0.173	0.154/0.129	0.000/0.000	0.042/0.026	0.042/0.023	0.318/0.083	0.260/0.200	0.000/0.000	0.394/0.310	0.054/0.036
Gemini-3.1-Pro	0.658/0.070	0.609/0.419	0.000/0.000	0.297/0.207	0.000/0.000	0.247/0.056	0.595/0.471	0.000/0.000	0.000/0.000	0.215/0.144
DeepSeek-V4	0.130/0.060	0.156/0.133	0.000/0.000	0.000/0.000	0.000/0.000	0.000/0.000	0.000/0.000	0.000/0.000	0.000/0.000	0.000/0.000
CliMB-DC†	0.000/0.000	0.000/0.000	0.000/0.000	0.000/0.000	–	–	–	–	–	–
HACHI†	0.000/0.000	0.000/0.000	0.000/0.000	0.000/0.000	–	–	–	–	–	–
HealthFlow*	0.792/0.089	0.958/0.792	0.681/0.029	0.810/0.611	–	–	–	–	–	–
MLZero*	0.253/0.023	0.222/0.170	0.000/0.000	0.000/0.000	–	–	–	–	–	–
MAGMA	0.827/0.247	0.897/0.655	0.697/0.026	0.920/0.856	0.834/0.639	0.863/0.407	0.786/0.436	0.840/0.899	0.676/0.411	0.723/0.425

Table S17: Effect of prompt template. The single-agent row group pools GPT-5.5, Claude-Opus-4.8, Gemini-3.1-Pro, and DeepSeek-V4 ($n = 120$ attempted runs per template: 4 systems $\times$ 10 case studies $\times$ 3 runs); MAGMA is evaluated over 10 case studies $\times$ 3 runs ($n = 30$ per template). HealthFlow, MLZero, CliMB-DC, and HACHI are evaluated on case studies 1–4 only, so each template cell covers $n = 12$ attempted runs and their rates are not comparable to the full grid. AUROC and AUPRC are averaged over completed runs only. Values are mean $\pm$ 1 standard deviation across runs; cells with a single completed run carry no standard deviation.

System	Template	Completed	Completion	AUROC	AUPRC
Single-agent (pooled)	Technical	22/120	18.3%	0.719 ±0.121	0.314 ±0.195
	Technical + Rubric	20/120	16.7%	0.759 ±0.123	0.413 ±0.233
	Clinical	37/120	30.8%	0.722 ±0.149	0.449 ±0.220
	Clinical + Rubric	29/120	24.2%	0.707 ±0.143	0.401 ±0.234
CliMB-DC*	Technical	0/12	0.0%	–	–
	Technical + Rubric	0/12	0.0%	–	–
	Clinical	0/12	0.0%	–	–
	Clinical + Rubric	0/12	0.0%	–	–
HACHI*	Technical	0/12	0.0%	–	–
	Technical + Rubric	0/12	0.0%	–	–
	Clinical	0/12	0.0%	–	–
	Clinical + Rubric	0/12	0.0%	–	–
HealthFlow*	Technical	11/12	91.7%	0.821 ±0.084	0.364 ±0.349
	Technical + Rubric	12/12	100.0%	0.836 ±0.101	0.434 ±0.340
	Clinical	12/12	100.0%	0.792 ±0.134	0.373 ±0.357
	Clinical + Rubric	12/12	100.0%	0.829 ±0.101	0.388 ±0.350
MLZero*	Technical	2/12	16.7%	0.887 ±0.002	0.677 ±0.006
	Technical + Rubric	1/12	8.3%	0.890	0.680
	Clinical	2/12	16.7%	0.757 ±0.008	0.068 ±0.005
	Clinical + Rubric	2/12	16.7%	0.763 ±0.000	0.071 ±0.000
MAGMA	Technical	30/30	100.0%	0.818 ±0.088	0.470 ±0.285
	Technical + Rubric	30/30	100.0%	0.804 ±0.126	0.531 ±0.318
	Clinical	30/30	100.0%	0.794 ±0.102	0.504 ±0.289
	Clinical + Rubric	30/30	100.0%	0.817 ±0.100	0.508 ±0.317

*Evaluated on case studies 1 - 4 only.

Table S18: Mean expert score per rubric item (0–2 scale), pooled over the available assessor scores, comparing the ten MAGMA packages with the sixteen baseline packages.

Rubric item	All packages	MAGMA	Baselines
1. Modality handling	1.79	1.75	1.81
2. Data cleansing and transformation	1.56	1.50	1.59
3. Feature engineering	1.46	1.50	1.44
4. Model comparison	0.48	**0.90**	0.22
5. Metric diversity	1.38	**1.55**	1.28
6. Optimisation	0.82	**1.32**	0.53
7. Scoring pipelines	1.85	1.70	1.94
8. Documentation	1.29	**1.70**	1.03

H PROMPT VARIANTS

To assess the robustness of MAGMA to differences in task formulation, each of the 10 case studies was evaluated using four prompt variants, resulting in 40 instantiated prompts. The variants follow a 2×2 design that varies (i) *clinical* versus *technical* task formulation and (ii) the absence versus presence of an explicit reporting rubric. The underlying prediction task, dataset, target outcome, and available data are held constant across the four variants. The templates below show the structure used to instantiate the prompts for each case study. The full prompts for each case study are provided in the supplementary material.

H.1 CLINICAL

The **Clinical** variant provides a concise description of the prediction problem from the perspective of a clinical user, without specifying detailed data-processing, modeling, or evaluation procedures.

> I have `{{DATA_CONTENT_SUMMARY}}` for `{{TARGET_POPULATION}}`. Based on this data, I want a system that can `{{TASK_DESCRIPTION}}`, so that `{{INTENDED_USE}}`.
>
> The data I can use to build this is at `{{EXPOSED_PATH}}`, and I have a separate set held back for testing later. `{{DATA_FORMAT_NOTES}}`
>
> Please provide the end-to-end Python code to build this for me and tell me how accurate it is.
>
> Please provide a final report summarizing the entire end-to-end process including key steps, metrics, and decisions made.
>
> Save everything to `{{OUTPUT_PATH}}`.

H.2 CLINICAL + RUBRIC

The **Clinical + Rubric** variant uses the same clinical formulation and appends the reporting rubric used in the evaluation.

> I have `{{DATA_CONTENT_SUMMARY}}` for `{{TARGET_POPULATION}}`. Based on this data, I want a system that can `{{TASK_DESCRIPTION}}`, so that `{{INTENDED_USE}}`.
>
> The data I can use to build this is at `{{EXPOSED_PATH}}`, and I have a separate set held back for testing later. `{{DATA_FORMAT_NOTES}}`
>
> Please provide the end-to-end Python code to build this for me and tell me how accurate it is.
>
> Please provide a final report summarizing the entire end-to-end process including key steps, metrics, and decisions made.
>
> Save everything to `{{OUTPUT_PATH}}`.
>
> **Please use this reporting rubric as a guideline:**
>
> `{{REPORTING_RUBRIC}}`

H.3 TECHNICAL

The **Technical** variant provides explicit information about the prediction task, dataset, preprocessing requirements, modeling guidance, evaluation metrics, and expected output artifacts.

> I need the code to train a model for `{{TASK_DESCRIPTION}}`. This is a binary classification task: `{{POSITIVE_CLASS_DEFINITION}}` versus `{{NEGATIVE_CLASS_DEFINITION}}`.
>
> The data comes from `{{DATASET_NAME}}` (`{{DATASET_BRIEF}}`), which contains `{{DATA_CONTENT_SUMMARY}}`. Clinically, this matters because `{{CLINICAL_RATIONALE}}`. The model is intended to `{{INTENDED_USE}}` for `{{TARGET_POPULATION}}`.
>
> The training data is at `{{EXPOSED_PATH}}`, and the model you produce will be tested on held-out evaluation data.
>
> `{{DATA_FORMAT_NOTES}}`

> Here's the data dictionary:
> {{DATA_DICT}}
> Before modelling, run the appropriate preprocessing and quality checks on this data. You need to handle the {{MODALITY}} the way it needs to be handled, check for the usual issues ({{DATA_QUALITY_CONCERNS}}), and apply whatever cleansing and transformation the data needs (normalisation/encoding/{{TRANSFORMATION_HINTS}}). Be explicit about how you're treating the predictors in terms of any rescaling, transformation, or standardisation you apply to {{KEY_PREDICTORS}}.
> For the model, {{MODEL_GUIDANCE}}. You may split the training data as you see fit and train then evaluate a model accordingly.
> Report performance with the metrics appropriate to this task {{PRIMARY_METRICS}} for {{METRIC_RATIONALE}}.
> Please provide the end-to-end Python code for this project. Save the trained model, the code, predictions on the held-out split ({{PATIENT_ID_COLUMN}}, predicted probability, predicted label), and a metrics file to {{OUTPUT_PATH}}.
> Please provide a final report summarizing the entire end-to-end process including key steps, metrics, and decisions made.

H.4 TECHNICAL + RUBRIC

The **Technical + Rubric** variant uses the same technical formulation and appends the reporting rubric used in the evaluation.

> *[Same template as the Technical variant above.]*
>
> **Please use this reporting rubric as a guideline:**
> {{REPORTING_RUBRIC}}

H.5 REPORTING RUBRIC

The same reporting rubric was appended to both rubric-based prompt variants. It covered five areas: **problem framing**, **data preparation**, **analytical methods**, **model specification**, and **model performance and evaluation**. The rubric included criteria related to preprocessing and quality checking, data leakage, patient-level partitioning, feature engineering, model selection and tuning, class imbalance, model specification, evaluation metrics, visualizations, and compute requirements.