MoBayes: A Modular Bayesian Framework for Separating Reasoning from Language in Conversational Clinical Decision Support

Yusuf Kesmen[*] LiGHT, EPFL
Fay Elhassan[*] LiGHT, EPFL
Jiayi Ma[*] University of Bern
Julien Stalhandske LiGHT, EPFL
Yena Chang LiGHT, EPFL
David Sasu LiGHT, EPFL
Alexandra Kulinkina LiGHT, EPFL
Akhil Arora[†] CLAN, Aarhus University
Lars Klein[†] LiGHT, EPFL
Mary-Anne Hartley[†] LiGHT, EPFL

Abstract

Large language models (LLMs) are increasingly used for conversational clinical decision support, yet they conflate next token prediction with probabilistic decision making. We argue that this conflation reflects an architectural limitation: such systems lack explicit posterior tracking, controllable abstention thresholds, and auditable reasoning chains. We introduce MoBayes, a **Mo**dular **Bayes**ian dialogue framework that separates reasoning from language. The LLM acts only as a language interface, parsing patient conversation into structured observations, while a Bayesian module performs probabilistic inference over these observations to update posteriors, select follow-up questions via expected-information-gain and determine when to stop or defer through calibrated decision thresholds. This design enables explicit posterior tracking, controllable selective decision-making, and replaceable population-specific statistical backends without retraining the language model. Across empirical and LLM-generated knowledge bases, MoBayes outperforms standalone frontier LLM doctors, including matched model-family comparisons where inexpensive sensor models paired with MoBayes exceed larger autonomous models at lower cost. The advantage persists under adversarial patient communication styles and across varying diagnostic scenarios. These results suggest that reliable conversational clinical decision support systems should separate probabilistic reasoning from language generation rather than scaling model size alone. Code is available at `https://github.com/EPFLiGHT/MoBayes`.

1 Introduction

Conversational clinical decision support (CCDS) is sequential abductive decision-making under uncertainty. Large language models are increasingly used in this role, yet they conflate two fundamentally different capabilities: natural-language communication and probabilistic inference, emulating abductive reasoning through next token prediction and token distributions rather than explicit posterior belief states. The result is systems that seem clinically fluent but lack the statistical grounding to make reliable decisions: no explicit probabilistic state, no principled information acquisition, uncalibrated confidence, and unauditable reasoning chains. The response is not better prompting or larger models

[*]Equal contribution. Corresponding author: `yusuf.kesmen@epfl.ch`
[†]Equal supervision

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

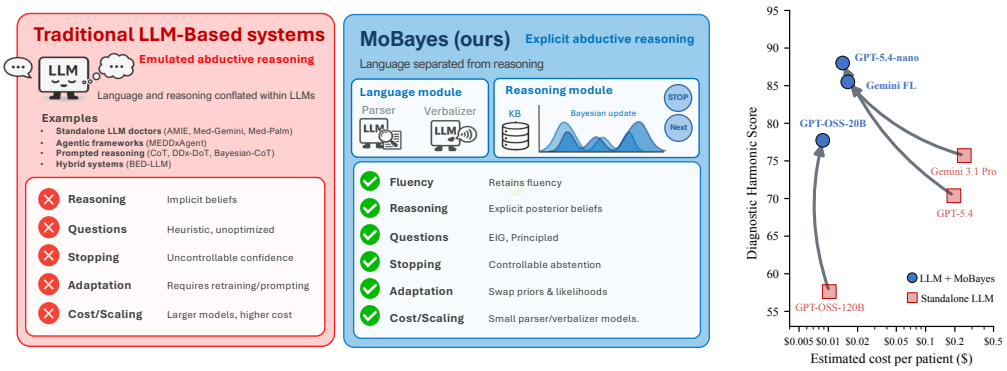


Figure 1: **(left)** Traditional LLM systems (Red) compared with MOBAYES (ours, blue). **(right)** Small sensor LLMs under MOBAYES beat larger same-family standalone models at lower cost. (DDxPlus 50, PatientSim.)

but *strict separation*: confine the LLM to language tasks and let a deterministic statistical model handle all probabilistic reasoning, with a tunable commit-or-defer rule.

Classical CDS demonstrated that structured rule based and probabilistic models can produce reliable clinical decisions (1; 2; 3; 4; 5) but failed to scale due to rigid non-conversational interfaces; modern LLMs (6; 7; 8) have largely solved the interface problem by enabling natural flexible interaction, but at the cost of abandoning explicit probabilistic rigor (no formal posterior, calibrated abstention mechanisms or auditable reasoning chain). Recent work injects probabilistic structure into LLM-driven pipelines (*probabilistic-LLM hybrids*) or wraps the LLM in multi-step orchestration (*agent-based diagnostic systems*) (9; 10; 11; 12; 13; 14; 15). However, these approaches still rely on LLMs for probability estimation, making them susceptible to hallucination and pseudo-probabilistic reasoning derived from token distributions rather than explicit disease modelling. Other approaches fine-tune LLMs on clinical data to internalise medical statistics (16), raising concerns about robustness, privacy, data governance, and auditability while also limiting sustainability: updates require retraining, which can distort previously learned distributions with no formal guarantee the model faithfully represents the underlying distributions at inference time.

We argue that the right response is not larger LLMs but *modular, statistically grounded reasoning*. We introduce **MoBayes**, a modular Bayesian framework that separates language from probabilistic reasoning. The LLM acts as a conversational interface, parsing patient interactions into structured observations and verbalizing questions selected by the Bayesian module. The Bayesian module performs inference over these observations, selects follow-up questions via expected information gain (EIG), and determines when to stop or defer using calibrated decision thresholds. These operations are defined over an explicit clinically grounded knowledge-base that can be adapted across populations without retraining the LLM, ensuring that no patient data is embedded in model weights.

Our main contributions are:

Framework. MOBAYES constrains the LLM to a structured–evidence interface and delegates inference, question selection, and abstention to an explicit Bayesian module.

Empirical advantage. Across three CCDS benchmarks, MoBayes outperforms prior baselines and matches or exceeds same-family frontier models at substantially lower per-token cost. The advantage persists under adversarial patient personas and across increasing diagnostic complexity.

Architectural attribution and selective control. Controlled comparisons with shared knowledge sources, including KBs elicited entirely from the same frontier LLM (§5.3), show that gains are architectural, not informational: the LLM is more usefully *elicited once* into a tabular KB and reasoned over by the Bayesian engine than invoked turn-by-turn as the reasoner. A single threshold τ further traces a continuous accuracy–coverage trade-off, enabling explicit control absent from standalone LLMs.

2 Related Work

Statistical clinical decision support. Early CDSS established that structured probabilistic and rule based reasoning produces clinically useful diagnoses, using formalisms ranging from certainty-factor rules and naïve Bayes to belief networks, and causal models (1; 17; 2; 3; 4; 5; 18). Despite strong accuracy, adoption stalled because these systems required structured pre-coded input and treated clinicians as data-entry operators, the "Greek Oracle" model (19). The bottleneck was *interfacing*, not reasoning, motivating designs that preserve auditable probabilistic reasoning while replacing the rigid interface.

End-to-end LLM-based diagnosis. Large language models substantially reduced the interfacing bottleneck by communicating in natural language. Med-PaLM 2 (6) and Med-Gemini (7) pushed standalone-LLM medical QA to expert level, and AMIE (8) extended this to multi-turn dialogue, with subsequent end-to-end variants pursuing similar designs (20; 15). These systems share a structural limitation: *no formal probabilistic state*, no explicit posterior, no information-theoretic question selection, and no auditable abstention. Conversation, question choice, diagnosis, and stopping are instead entangled into a single generative model. The absence of structured reasoning carries safety costs: clinical LLMs exhibit poor calibration (21), produce unstable reasoning across runs (22), are vulnerable to adversarial hallucination (23), and encode geographic and racial biases (24; 25).

Probability–language hybrids. A growing line of work augments LLM-driven pipelines with probabilistic or structured-reasoning components, including Bayesian experimental design, decision theory, active-learning planners, information-pursuit search, abstention-aware questioning, multi-agent decomposition, and structured reasoning (26; 9; 10; 11; 12; 27; 28; 13; 14; 29; 30). In all of these systems the LLM remains *inside* the diagnostic decision loop, as probability source, candidate generator, planner substrate, or latent reasoning state. An orthogonal line fine-tunes LLMs on clinical data (31; 16), but learned distributions are not directly auditable and raise data-governance and privacy concerns.

Our position: strict separation. We propose *strict separation* between language and clinical reasoning: the LLM handles *only* language (parsing utterances into structured evidence and verbalising questions selected by the Bayesian module), while all belief updates, question selection, stopping, and abstention are delegated to an external, inspectable statistical engine. This separates the two components that prior work has historically traded off: classical CDSS-style auditable statistics and LLM-style conversational interface. The diagnostic state remains outside the LLM, patient data is never used for training or fine-tuning, and adapting to a new population requires only updating priors and likelihoods rather than re-aligning the LLM. Figure 1a illustrates this taxonomy; Appendix D expands on each of the families above and provides a system-by-system comparison in Table 25.

3 Method

We formulate diagnostic dialogue as a sequential Bayesian decision process. Let $\mathcal{D} = \{d_1, \ldots, d_K\}$ denote the set of K candidate diseases and $\mathcal{F} = \{f_1, \ldots, f_N\}$ the set of N observable clinical features, where each feature f_i takes values in a finite set $\mathcal{V}_{f_i}$. At each turn t, the system maintains a posterior belief $\mathbf{b}_t \in \Delta^{K-1}$ over diseases, selects the most informative feature to query, updates its belief from the patient's response, and decides whether to terminate. The objective is to commit to a diagnosis when the posterior crosses a threshold τ and abstain otherwise, within the fixed question budget $T_{\max}$.

MOBAYES decomposes clinical dialogue into two modular components: a *language interface* and a *Bayesian reasoning module*. The LLM acts only as a language layer, parsing patient utterances and verbalizing questions selected by the Bayesian module. The Bayesian module maintains posterior beliefs, selects questions, and renders clinical decisions. The two communicate exclusively through structured evidence triples (f, v, c): feature identifier, schema-validated value, and confidence weight. All probabilistic reasoning is performed by the Baysian module, yielding a fully auditable decision trace. The complete procedure is given as Algorithm 1 in Appendix B.1.

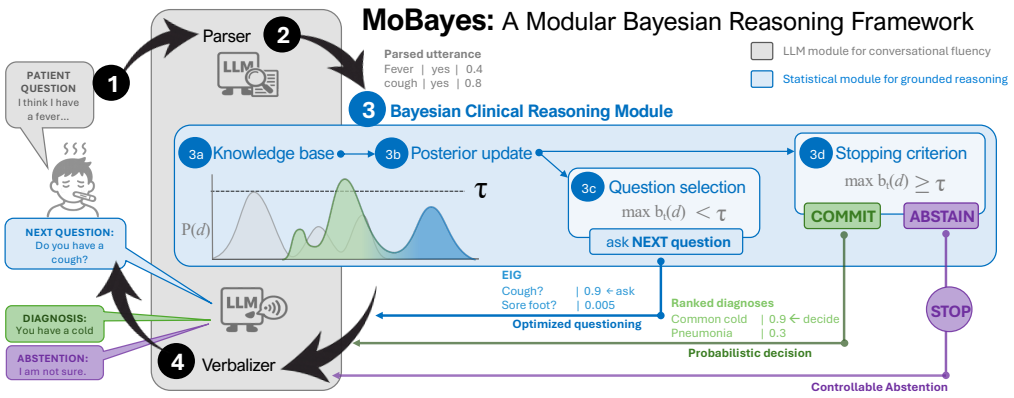


Figure 2: Overview of the MOBAYES architecture. The LLM acts only as a language layer (parses utterances into evidence triples (f, v, c), verbalises questions selected by the Bayesian module); all reasoning, posterior updates, EIG-based question selection, stopping, and controllable abstention reside in the deterministic, auditable Bayesian module.

3.1 Knowledge base

The Bayesian module reasons over a tabular knowledge base of disease priors $\boldsymbol{\pi}$ and per-feature conditional likelihoods $P(X_f{=}v \mid d)$. We use two interchangeable construction paths that populate the same schema.

Empirical KB (training data). On DDxPlus we accumulate co-occurrence counts from the published 1.03M training records (32) and convert them into Dirichlet–Categorical likelihoods (full formulation in Appendix B.4, Eq. 12). The 49 pathologies and 314 features are taken directly from the dataset.

LLM-elicited KB. When labelled training data is unavailable, the same KB schema is populated by structured prompting of a frontier LLM in two stages, first feature catalogues per disease, then per-feature conditional distributions, with deduplication and validation. The motivation is modular separation: asking the LLM to externalise its knowledge into an explicit, auditable KB and perform inference separately through Bayesian updating. Two LLM-elicited KBs are used in this paper: (i) for the cross-KB study we draw the disease list from the BMRB Genes/Disease catalogue (33) restricted to the *respiratory diseases* (6) and *acquired diseases* (12) categories, yielding 18 diseases with a deliberate mix of symptomatically overlapping and clearly distinct conditions; (ii) for the AgentClinic-MedQA column we keep AgentClinic's disease list and elicit features and conditional likelihoods over it. Full prompts and validation rules are in Appendix B.4.

3.2 Bayesian reasoning module

Let $\mathcal{C} = \{c_1, \ldots, c_5\} \subset (0, 1]$ denote the set of confidence weights and $\phi\colon \mathcal{L} \to \mathcal{C}$ the mapping from linguistic labels $\mathcal{L} = \{\textit{very_likely}, \textit{likely}, \textit{uncertain}, \textit{unlikely}, \textit{very_unlikely}\}$ to numerical weights. At each turn t, the patient's response about feature f_t is parsed into a tuple $(v_t, c_t) \in \mathcal{V}_{f_t} \times \mathcal{C}$.

Soft evidence and belief update. Patient responses often hedge ("I think I had a fever"). Rather than treating every observation as hard evidence, we use a Pearl-style soft-evidence update (34; 35): a parser-supplied confidence $c \in (0, 1]$ mixes the standard likelihood with a neutral vector. At $c{=}1$ we recover the standard Bayesian update; at $c{\to}0$ the evidence has vanishing effect. The five linguistic confidence labels and their numerical weights, the log-space normalisation, and the closed-form update equations are deferred to Appendix A.1 (Eqs. 3–4). For numeric features a Gaussian-weighted soft match replaces the point likelihood before the soft-evidence update is applied.

Question selection. At each turn the Bayesian module queries the unasked feature with the highest expected information gain (EIG), the standard myopic Bayes-optimal policy. We adopt the naive-Bayes factorisation $X_f \perp \mathbf{e}_{1:t} \mid D$; richer factorisations (tree-augmented naive Bayes (36), noisy-OR (34), latent comorbidity factors (37), sum-product networks (38)) plug in without altering belief

representation, EIG, or abstention, at the cost of more complex inference and a quadratic blow-up in the LLM-querying budget for KB construction. Per-turn cost is linear in $|\mathcal{F}|$, $\max_f |\mathcal{V}_f|$, and K. Closed forms for EIG, the counterfactual posterior, computational cost, and policy ablations (hierarchical EIG, top-k focused EIG, confidence-activated hybrid) are in Appendix A.2 (Eqs. 7–10).

3.3 Orchestration: ending the consultation

The system must decide not only *what* to diagnose but *whether* to diagnose at all. After each belief update we evaluate stopping conditions in strict priority: a warm-up period of $T_{\min}$ turns prevents premature commitment while early posteriors remain dominated by the prior; once warm-up elapses, the system commits to $d^* = \arg\max_d b_t(d)$ as soon as $\max_d b_t(d) \geq \tau$; a hard budget $T_{\max}$ guarantees termination, abstaining if no disease has crossed τ by then (budget values are in Appendix A.3). The threshold τ is itself a deployment parameter that gives providers explicit control over the accuracy–coverage trade-off, from high-throughput triage (τ low, high coverage) to safety-critical referral (τ high, near-zero error on committed cases), without retraining; autonomous LLM doctors offer no equivalent mechanism, and we characterise this trade-off in Section 5.5.

3.4 Patient-facing interface

The LLM serves as a pure language layer with no access to the posterior distribution or diagnostic state (prompt templates in Appendix B.2).

Response parsing. Given a patient utterance u_t and the queried feature f_t with schema $\mathcal{S}_{f_t} = (\mathcal{V}_{f_t}, \mathcal{C})$, the parser produces $(v_t, c_t) \in \mathcal{V}_{f_t} \times \mathcal{C}$ via a three-tier cascade. First, *deterministic pattern matching* maps clear affirmative, negative, and hedging expressions to (v, c) pairs at zero API cost. When this tier abstains on ambiguous input, an *LLM-based extractor* classifies the response into the schema's value set with a confidence label from $\mathcal{L}$, mapped to a numerical weight via ϕ. The key design choice is that the LLM performs *structured classification* into $\mathcal{V}_f \times \mathcal{L}$, a small, fixed label set, rather than continuous probability estimation. LLMs exhibit strong ordinal judgement (10) but poor calibration for continuous probabilities (39); the discrete-to-numerical mapping ϕ shifts the calibration burden from continuous probability estimation to a coarser ordinal task that LLMs handle more reliably. From the Bayesian module's perspective, each (v, c) pair is a noisy observation, and the LLM functions as a discretized probabilistic model rather than a reasoning system.

Question verbalisation. Given the EIG-selected feature f^*, the LLM receives its name, answer schema $\mathcal{S}_{f^*}$, and a coarse confidence indicator that modulates conversational tone only; the LLM never receives $\mathbf{b}_t$, $H(\mathbf{b}_t)$, or any other internal model state.

4 Experimental Setup

4.1 Datasets

We evaluate on three diagnostic-dialogue benchmarks. **DDxPlus** (32) is a large-scale synthetic dataset of $\sim$1.3M patient records over 49 diseases and 314 features with a published train/test split; the empirical KB is built from the training split and evaluation uses held-out test patients. To isolate architectural from informational advantages, we additionally evaluate on an **LLM-elicited KB** whose likelihoods are elicited *entirely* from a frontier LLM (Section 3.1) in two ecosystems (GPT-5.4 and Gemini 3.1) over a disease list drawn from the BMRB Genes/Disease catalogue (33) restricted to the respiratory and acquired-diseases categories; the same LLM also serves as the standalone doctor in our fair-fight experiments. **AgentClinic-MedQA** (40) is a USMLE-derived multi-agent dialogue benchmark in which the doctor LLM is evaluated through interaction with the suite's own patient, measurement, and moderator agents. KB construction, case curation, and elicitation protocols are detailed in Appendices B.4, C.14, and B.4.

4.2 Patient simulation

Evaluating a diagnostic dialogue system requires simulated patients that respond realistically to follow-up questions; several frameworks address this challenge (41; 42; 43; 44). We use three

published patient simulators across our experiments, each corresponding to a different benchmark setting.

PatientSim. For the DDxPlus PatientSim column we adopt the PatientSim simulator (43) with minor interface-level adaptations to plug it into our DDxPlus profile schema; the persona taxonomy (personality, language proficiency, recall accuracy, cognitive state) is unchanged. Main experiments use the cooperative *plain* persona; adversarial-persona results are in Appendix C.8.

MEDDxAgent history-taking simulator. For the second DDxPlus column we use MEDDxAgent's published history-taking simulator (14) on a separate DDxPlus subset, with their orchestrator (DDxDriver) and bench class invoked from their open-source codebase unmodified.

AgentClinic patient agent. As an orthogonal, non-DDxPlus column, we use AgentClinic's published patient and measurement components (40) on AgentClinic-MedQA, with their dialogue protocol and native moderator unchanged.

Each simulator receives the ground-truth profile and never accesses the Bayesian module or KB. Our adaptations are interface-level only and preserve each simulator's dialogue protocol; details in Appendix B.5.

4.3 Models and baselines

We compare MOBAYES against three families of prior diagnostic dialogue systems, all run under the same patient simulator and disease universe per benchmark column. Full specifications, prompts, and per-method hyperparameters are in Appendix C.14; Table 6 lists the LLM backbones.

MOBAYES variants. Each variant pairs the Bayesian engine with a different LLM sensor; since the LLM only parses and verbalises, no medical specialisation is needed. We test six inexpensive backends across providers and architectures (Table 6); only the sensor LLM changes between variants.

Standalone frontier LLM doctors. The standalone baseline gives the LLM full control over the clinical dialogue: it decides what to ask, when to stop, and what to diagnose, with no external reasoning aid. We evaluate six frontier models (Table 6); the prompt is in Appendix B.2 and the closed-set normalisation protocol used to score free-text predictions in Appendix B.3.

Prior diagnostic-dialogue systems. We evaluate published systems spanning the four families surveyed in Section 2: *prompt-based* (Chain-of-Thought (29), DDx-CoT and Bayesian-CoT (45)); *information-pursuit* (UoT (28)); *agent-based* (MediQ (13), which is MCQ-bound; we split each benchmark into two disjoint subsets to keep its option pool tractable, and for fairness all other baselines are run on the same two-subset partition; MEDDxAgent (14) in its paper-best configuration; AgentClinic's own doctor agent (40)); and a *fine-tuned* reference (DiagnosisGPT-34B / Chain-of-Diagnosis (31)). We additionally include three configurations of our own to round out the comparison: a *Closed-World* prompt-based baseline, a *strict-abstain* scoring variant of MediQ, and a *closed-world ablation* of MEDDxAgent. Per-method hyperparameters and the rationale for each added configuration are detailed in Appendix C.14. Every non-fine-tuned baseline runs with the same gpt-5.4-nano backbone we use for our headline MOBAYES sensor, so any per-column differences isolate architectural rather than backbone effects.

4.4 Evaluation metrics

Following the selective prediction framework (46), a selection function $g(x) \in \{0, 1\}$ decides whether to commit (g=1) or abstain (g=0) based on posterior confidence exceeding τ. We report two families of metrics: **Top-k accuracy** ($k \in \{1, 3\}$) measures diagnostic correctness over all patients, counting abstentions as errors; and the *commit-or-abstain* **selective diagnosis** metrics defined below.

Selective diagnosis metrics. The *commit-or-abstain* trade-off is summarised by selective accuracy, coverage, and the **Diagnostic Harmonic Score (DHS)**:

$$\text{Sel. Acc} = \frac{\sum_i g(x_i)\mathbb{1}[f(x_i)=y_i]}{\sum_i g(x_i)}, \quad \text{Cov} = \frac{1}{n}\sum_i g(x_i), \quad \text{DHS}_\alpha = \left(\frac{\alpha}{\text{Sel. Acc}} + \frac{1-\alpha}{\text{Cov}}\right)^{-1}. \tag{1}$$

Table 1: Combined n=100 comparison of MOBAYES vs prior diagnostic-dialogue systems. Rows ranked within each category by average DHS. MOBAYES entries marked * are statistically significant (paired bootstrap, $p<0.05$) over the strongest non-MOBAYES baseline in the same column. Per-cell 95% bootstrap CIs in Appendix Table 20. AgentClinic-MedQA uses a GPT-5.4-nano-elicited KB (Appendix B.4). DHS coloured per column: the **greener** the **better**, the **redder** the **worse**.

	DDxPlus 100 (MOBAYES PatientSim)				DDxPlus 100 (HT-sim)				AgentClinic-MedQA 100			
Method	T-1	SA	DHS ↑	Tok/case	T-1	SA	DHS ↑	Tok/case	T-1	SA	DHS ↑	Tok/case
Prompt-based												
DDx-CoT	42	51	55	28.8K	37	78	16	18.5K	40	78	62	23.7K
Closed-World (CW)	40	50	55	32.0K	45	75	21	23.9K	40	75	62	25.7K
Chain-of-Thought (CoT)	52	55	69	30.8K	41	63	29	15.8K	71	**89**	84	22.5K
Bayesian-CoT	60	62	74	46.6K	41	58	45	34.1K	63	77	79	37.8K
Information-pursuit												
UoT	28	28	44	37.7K	5	5	10	28.2K	38	40	56	34.6K
Agent-based												
MediQ-Expert (strict abstain)	32	51	56	39.3K	25	57	50	41.9K	58	85	76	30.0K
AgentClinic	27	51	52	18.1K	28	44	52	29.7K	66	67	80	6.2K
MediQ-Expert (force-final)	43	43	60	39.3K	34	34	51	41.9K	76	76	86	30.0K
MEDDxAgent (closed-world abl.)	41	41	58	79.9K	39	39	56	71.9K	**80**	80	**89**	69.0K
MEDDxAgent (paper-best)	45	45	62	108.3K	56	56	72	88.1K	71	71	83	106.8K
Fine-tuned (specialised model)												
CoD / DiagnosisGPT-34B	11	11	20	13.4K	4	6	11	12.7K	46	78	67	31.2K
Ours												
MOBAYES (gpt-5.4-nano)	**82**	**85**	**90***	16.4K	**81**	**87**	**90***	18.0K	70	74	83	18.9K

At α=0.5 (used throughout) DHS is the harmonic mean of SelAcc and Coverage; the limits $\alpha\to1$ and $\alpha\to0$ recover SelAcc and Coverage respectively. τ is a *deployment parameter*: providers shift along the accuracy–coverage curve to match clinical risk tolerance (Section 5.5).

5 Results

5.1 Comparison with prior diagnostic-dialogue systems

Table 1 reports the head-to-head comparison against the baselines of Section 4.3. MOBAYES achieves the highest DHS on both DDxPlus columns and remains competitive with the best agent-based baseline on AgentClinic-MedQA, while non-MOBAYES methods vary substantially across columns and the fine-tuned reference (DiagnosisGPT-34B) underperforms across all columns.

5.2 Comparison with frontier standalone LLMs

Table 2 compares six MOBAYES model variants against six frontier standalone LLM doctors across all three diagnostic-dialogue benchmarks. Every MOBAYES variant outperforms every standalone doctor on DHS on at least two of the three benchmarks, and the headline variant (MOBAYES + GPT-5.4-nano) leads on all three. The best MOBAYES variant achieves the highest DHS at a fraction of the per-token cost of the best standalone doctor. The advantage is architectural: every MOBAYES variant achieves higher point-estimate DHS than every standalone doctor on DDxPlus, and the headline best-vs-best gap is large.

5.3 LLM-generated knowledge base

A natural objection is that MOBAYES's advantage stems from access to data-derived statistics unavailable to the standalone doctor. To control for this, we replace the empirical DDxPlus KB with ones generated entirely by LLMs. In each ecosystem the same frontier LLM populates the KB and, in a separate elicitation, generates evaluation patients (Appendix B.4); both are noisy realisations of the LLM's beliefs. An inexpensive same-family MOBAYES sensor matches the frontier standalone's

Table 2: MoBayes sensors vs. standalone LLM doctors on the three benchmarks of Section 4.1; n=50 per cell. MoBayes uses DHS-optimal τ^* per sensor (Appendix A.4). DHS coloured per benchmark column. Per-cell 95% bootstrap CIs are reported in Appendix Table 19; the AgentClinic-MedQA results use a knowledge base elicited from GPT-5.4-nano (construction protocol in Appendix B.4).

	DDxPlus 50 (PatientSim)			**DDxPlus 50 (HT-sim)**			**AgentClinic-MedQA 50**		
System	T-1	SA	DHS ↑	T-1	SA	DHS ↑	T-1	SA	DHS ↑
Standalone LLM doctors (frontier)									
SA GPT-OSS-120B	40	40.8	57.6	54	54.5	59.7	48	49.0	65.3
SA Kimi K2.5	42	54.1	62.5	42	40.7	46.4	54	57.4	71.3
SA Qwen 3.6 Plus	48	51.1	66.2	60	65.0	71.7	64	63.3	76.9
SA Llama-4-Maverick	56	56.5	70.0	46	56.0	52.8	56	59.6	72.9
SA GPT-5.4	60	64.1	70.4	58	76.0	60.3	60	62.8	72.6
SA Gemini 3.1 Pro	60	62.5	75.7	60	73.2	77.3	66	70.2	80.4
MoBayes + *sensor LLM (ours)*									
+ Llama-4-Scout	56	60.9	73.3	80	**87.0**	89.4	68	79.1	82.4
+ Gemma 4 31B	58	61.7	74.5	82	83.7	90.3	64	65.3	78.4
+ MiniMax M2.5	60	66.7	76.6	60	75.0	77.4	50	52.1	67.5
+ GPT-OSS-20B	60	68.2	76.8	50	61.0	69.9	50	58.1	69.4
+ Gemini 3.1 Flash Lite	74	77.1	85.5	**84**	85.7	**91.4**	**78**	**86.7**	**88.3**
+ GPT-5.4-nano	**78**	**81.2**	**88.0**	80	83.3	89.2	76	79.2	86.8

Sel. Acc in both ecosystems at roughly 10x lower per-token cost, and the standalone drops sharply on rare diseases drawn from its own world model (Appendix C.1).

5.4 Cost-effectiveness

A key advantage of MoBayes is that the Bayesian module performs all diagnostic reasoning deterministically at zero API cost; the LLM handles only parsing and verbalisation. Figure 3 (left) shows DHS vs. per-token API pricing across model scales: MoBayes models cluster in the top-left quadrant (cheap and accurate) while frontier standalone doctors occupy the bottom-right. The corresponding per-patient cost view, which folds in dialogue length and per-turn token routing, is reported in Figure 10 of Appendix C.7. Figure 1b isolates this effect within three model families. Each arrow shows the result of replacing a frontier standalone doctor with a cheap MoBayes models from the same family; every arrow points towards lower cost and higher DHS.

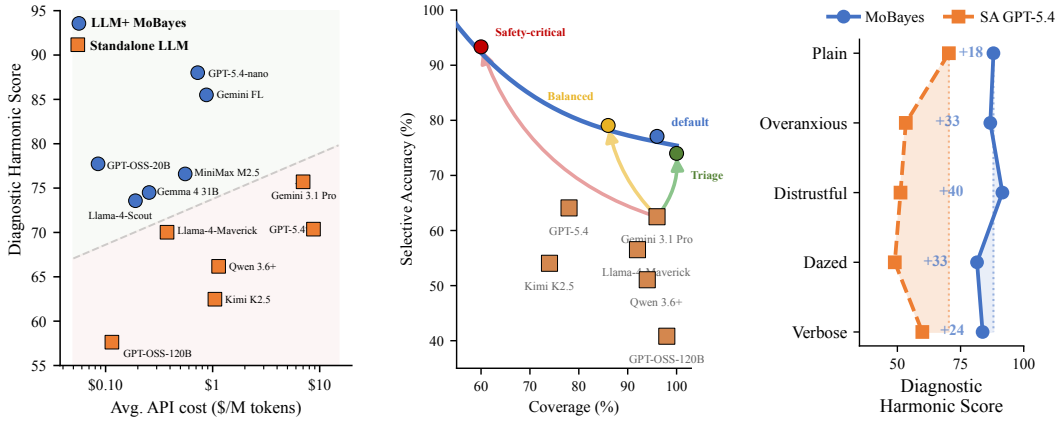


Figure 3: **(a) Cost–accuracy tradeoff.** DHS vs. per-token cost: MoBayes models (circles) outperform standalone doctors (squares) at lower cost (Section 5.4). **(b) Controllable selective diagnosis.** Sweeping τ traces the accuracy–coverage frontier for MoBayes + Gemini FL; arrows mark deployment modes against SA Gemini 3.1 Pro (Section 5.5). **(c) Persona robustness.** DHS across five PatientSim personas; shaded bands mark degradation from *Plain* baseline (Appendix C.8). (DDxPlus 50, PatientSim.)

5.5 Controllable selective diagnosis

A key property of MOBAYES is that deployers can choose their operating point along the accuracy, coverage curve by adjusting τ. Figure 3 (right) illustrates the contrast: MOBAYES traces a continuous curve as τ varies, while standalone doctors (×) are locked at a single, uncontrollable point. The labelled modes mark representative deployment points. This controllability corresponds to shifting the emphasis parameter α in Eq. 1: α towards 1 penalises errors (safety-critical referral), while α towards 0 penalises abstention (high-throughput triage). Standalone LLMs offer no equivalent: they either diagnose or abstain based on the model's internal, uncalibrated confidence.

5.6 Robustness across patient personas

A clinically deployed system must remain reliable under noisy, evasive, or confused patient communication. Figure 3c compares DHS across five PatientSim personas: standalone GPT-5.4 collapses sharply under adversarial styles (distrustful, dazed, verbose, overanxious), while MOBAYES stays nearly stable because the Bayesian backbone accumulates evidence independently of the patient's surface communication style (full breakdown in Appendix C.8).

5.7 Additional results

We report further experiments in the appendix: confidence threshold tuning and accuracy, coverage trade-offs (Appendices A.4, C.6), question-selection policy ablations (Appendix C.4), prior distribution sensitivity (Appendix C.5), and disease scaling analysis examining how performance varies with the number of candidate diseases (Appendix C.11). Appendices C.15 to C.20 include additional analyses.

6 Discussion

Why strict separation works. Diagnosis is fundamentally an abductive probabilistic problem: sequentially updating competing disease hypotheses as evidence accumulates. LLMs lack persistent belief state, cannot compute information-theoretic objectives, and produce uncalibrated confidence through implicit token distributions rather than explicit posterior inference. By delegating these tasks to a deterministic Bayesian engine, MOBAYES gains exact posterior tracking, principled question selection, and controllable abstention. The thesis is supported across multiple experiments: MOBAYES achieves the highest DHS on both DDxPlus columns of Table 1, remains competitive on AgentClinic-MedQA, and matches or exceeds same-family frontier standalone LLMs, including in the fair-fight setting where both draw from the same LLM's beliefs (Appendix B.4). The modular architectural choice also carries into cost-effectiveness: MOBAYES consumes notably few tokens per session, matching or exceeding same-family frontier standalones at a fraction of the per-session cost (Figure 3; breakdown in Appendix C.7), with within-family pairings consistently in our favour. Standalone LLMs produce a single take-it-or-leave-it diagnosis. MOBAYES traces a continuous accuracy–coverage frontier via a single threshold τ (Section 5.5), with no retraining. The Bayesian backbone is also notably robust to adversarial patient communication styles where standalone doctors collapse: under PatientSim's distrustful, dazed, and overanxious personas, MOBAYES maintains stable DHS while the standalone DHS drops sharply (Appendix C.8). The same robustness extends to low-prevalence diseases (Appendix C.9) and to larger candidate disease spaces (Appendix C.11).

Limitations. MOBAYES operates under a *closed-world assumption*: it can only diagnose diseases present in the knowledge base, whereas standalone LLMs can draw on open-ended medical knowledge. KB coverage is therefore a critical deployment concern, though the abstention mechanism provides a safety valve and the LLM-elicited KB experiment (Section 5.3) shows that constructing a KB does not require labelled clinical data. We also rely on simulated patients rather than real clinical interactions, and the current architecture cannot exploit volunteered evidence that falls outside the asked question, a behaviour standalone LLMs handle naturally by reading the full conversation. The closed world is also a clinical boundary, since a fixed candidate list cannot capture unexpected or rare diseases, or several at once. Choosing which diseases enter that list is itself a clinical decision. When the true diagnosis is absent, MOBAYES mostly abstains, but it cannot name the missing disease (Appendix C.15). Simulator-based results are not evidence of clinical readiness. Prospective

evaluation with real patients and clinicians is required before any claim of clinical usefulness. Results depend on the quality of the KB, so KB swapping requires reliable local estimation, validation, and maintenance.

Broader impact. MOBAYES is designed for clinical decision *support*, rather than autonomous diagnosis: the abstention mechanism declines uncertain cases rather than forcing a position, and the full decision trace (prior, evidence, posterior per turn) is auditable. The LLM serves only as a language interface and is never trained on patient data, so no patient information is embedded in model weights. Deploying in a new population requires only a local KB, not retraining, making the framework practical for low-resource settings where clinical data exists but compute for LLM fine-tuning does not.

7 Conclusion

We presented MOBAYES, a diagnostic dialogue framework built on a strict separation principle: an LLM handles only language, while a deterministic Bayesian module handles all probabilistic reasoning over an explicit knowledge base. The architecture is auditable, architecturally private, and deployable in low-resource settings: swapping the KB adapts the system to a new population with no retraining of the language model. Across multiple benchmarks, MOBAYES matches or exceeds diagnostic-dialogue baselines and same-family frontier standalone LLMs at a small fraction of the cost. The architectural origin of the advantage is supported by a control setting in which MOBAYES and the standalone doctor draw clinical knowledge from the same frontier LLM, yet MOBAYES still outperforms the standalone system. Our broader conclusion is that reliable conversational clinical decision support should not rely on implicit reasoning from next-token prediction alone: LLMs for communication, explicit probabilistic systems for calibrated reasoning.

Acknowledgments and Disclosure of Funding

This work was supported by the EPFL AI Center PhD Fellowship Program 2026. The LiGHT lab is partly supported by the Gates Foundation (INV-076674), the Kristian Gerhard Jebsen Foundation, Swisscom (Schweiz) AG through the Swiss National AI Institute Industry Partnership Program, and the ETH Domain through the Swiss AI Initiative. Support from the Swiss AI Initiative includes computational resources on the Swiss National Supercomputing Centre (CSCS) Alps infrastructure (project IDs 27 and a0238) and a PhD fellowship for the first author. Arora's lab is partly supported by grants from the Novo Nordisk Foundation (NNF24OC0099109), the Pioneer Centre for AI, EU Horizon 2020 (101168951), and Danish Advanced Research Academy (DARA) and Danish Data Science Academy (DDSA) PhD Fellowships. We also gratefully acknowledge generous gifts from Microsoft and IT-vest networking universities. The authors declare no competing interests.

References

[1] Edward H. Shortliffe. *Computer-Based Medical Consultations: MYCIN*. Elsevier, 1976.

[2] F. T. de Dombal, D. J. Leaper, J. R. Staniland, A. P. McCann, and Jane C. Horrocks. Computer-aided diagnosis of acute abdominal pain. *British Medical Journal*, 2(5804):9–13, 1972.

[3] Randolph A. Miller, Harry E. Pople, and Jack D. Myers. Internist-I, an experimental computer-based diagnostic consultant for general internal medicine. *New England Journal of Medicine*, 307(8):468–476, 1982.

[4] G. Octo Barnett, James J. Cimino, Jon A. Hupp, and Edward P. Hoffer. DXplain: An evolving diagnostic decision-support system. *JAMA*, 258(1):67–74, 1987.

[5] David E. Heckerman, Eric J. Horvitz, and Bharat N. Nathwani. Toward normative expert systems: Part I. The Pathfinder project. *Methods of Information in Medicine*, 31(2):90–105, 1992.

[6] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, et al. Toward expert-level medical question answering with large language models. *Nature Medicine*, 31:943–950, 2025. doi: 10.1038/s41591-024-03423-7. URL https://www.nature.com/articles/s41591-024-03423-7.

[7] Khaled Saab, Tao Tu, Wei-Hung Weng, Ryutaro Tanno, David Stutz, Ellery Wulczyn, et al. Capabilities of Gemini models in medicine. *arXiv preprint arXiv:2404.18416*, 2024. URL https://arxiv.org/abs/2404.18416.

[8] Tao Tu, Mike Schaekermann, Anil Palepu, et al. Towards Conversational Diagnostic Artificial Intelligence. *Nature*, 642:442–450, 2025. doi: 10.1038/s41586-025-08866-7. URL https://www.nature.com/articles/s41586-025-08866-7.

[9] Deepro Choudhury, Sinead Williamson, Adam Goliński, Ning Miao, Freddie Bickford Smith, Michael Kirchhof, Yizhe Zhang, and Tom Rainforth. BED-LLM: Intelligent information gathering with LLMs and Bayesian experimental design. In *Proceedings of the Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=qyylZMLYT8.

[10] Ollie Liu, Deqing Fu, Dani Yogatama, and Willie Neiswanger. DeLLMa: Decision making under uncertainty with large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=Acvo2RGSCy. Spotlight.

[11] Yu Feng, Ben Zhou, Weidong Lin, and Dan Roth. BIRD: A trustworthy Bayesian inference framework for large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=fAAaT826Vv. Oral.

[12] Jiayuan Zhu, Jiazhen Pan, Yuyuan Liu, Fenglin Liu, and Junde Wu. Ask patients with patience: Enabling LLMs for human-centric medical dialogue with grounded reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2846–2857, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.142. URL https://aclanthology.org/2025.emnlp-main.142/.

[13] Shuyue Stella Li, Vidhisha Balachandran, Shangbin Feng, Jonathan S. Ilgen, Emma Pierson, Pang Wei Koh, and Yulia Tsvetkov. MediQ: Question-asking LLMs and a benchmark for reliable interactive clinical reasoning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0908. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/32b80425554e081204e5988ab1c97e9a-Abstract-Conference.html.

[14] Daniel Philip Rose, Chia-Chien Hung, Marco Lepri, Israa Alqassem, Kiril Gashteovski, and Carolin Lawrence. MEDDxAgent: A unified modular agent framework for explainable automatic differential diagnosis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13803–13826, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.677.

[15] Xinyi Liu, Dachun Sun, Yi R. Fung, Dilek Hakkani-Tür, and Tarek F. Abdelzaher. DocCHA: Towards LLM-augmented interactive online diagnosis system. In *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDial)*, pages 609–619, Avignon, France, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.sigdial-1.49/.

[16] Minkyoung Kim, Yunha Kim, Hee Jun Kang, Hyeram Seo, Heejung Choi, JiYe Han, Gaeun Kee, Seohyun Park, Soyoung Ko, HyoJe Jung, Byeolhee Kim, Tae Joon Jun, and Young-Hak Kim. Fine-tuning LLMs with medical data: Can safety be ensured? *NEJM AI*, 2(1), 2025. doi: 10.1056/AIcs2400390.

[17] Victor L. Yu, Lawrence M. Fagan, Sharon M. Wraith, William J. Clancey, A. Carlisle Scott, John Hannigan, Robert L. Blum, Bruce G. Buchanan, and Stanley N. Cohen. Antimicrobial selection by a computer: A blinded evaluation by infectious diseases experts. *JAMA*, 242(12): 1279–1282, 1979.

[18] Sholom M. Weiss, Casimir A. Kulikowski, Saul Amarel, and Aran Safir. A model-based method for computer-aided medical decision-making. *Artificial Intelligence*, 11(1–2):145–172, 1978.

[19] Randolph A. Miller and Fred E. Masarie. The demise of the "Greek Oracle" model for medical diagnostic systems. *Methods of Information in Medicine*, 29(01):1–2, 1990.

[20] Kaishuai Xu, Yi Cheng, Wenjun Hou, Qiaoyu Tan, and Wenjie Li. Reasoning like a doctor: Improving medical dialogue systems via diagnostic reasoning process alignment. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6796–6814, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.406. URL https://aclanthology.org/2024.findings-acl.406/.

[21] Thomas Savage, John Wang, Robert Gallo, Abdessalem Boukil, Vishwesh Patel, Seyed Amir Ahmad Safavi-Naini, Ali Soroush, and Jonathan H. Chen. Large language model uncertainty proxies: Discrimination and calibration for medical diagnosis and treatment. *Journal of the American Medical Informatics Association*, 32(1):139–149, 2025. doi: 10.1093/jamia/ocae254.

[22] Nearchos Potamitis, Lars Klein, and Akhil Arora. ReasonBENCH: Benchmarking the (in)stability of LLM reasoning. *arXiv preprint arXiv:2512.07795*, 2025.

[23] Mahmud Omar, Vera Sorin, Jeremy D. Collins, David Reich, Robert Freeman, Nicholas Gavin, Alexander Charney, Lisa Stump, Nicola Luigi Bragazzi, Girish N. Nadkarni, and Eyal Klang. Multi-model assurance analysis showing large language models are highly vulnerable to adversarial hallucination attacks during clinical decision support. *Communications Medicine*, 5(1):330, 2025. doi: 10.1038/s43856-025-01021-3. URL https://www.nature.com/articles/s43856-025-01021-3.

[24] Jesutofunmi A. Omiye, Jenna C. Lester, Simon Spichak, Veronica Rotemberg, and Roxana Daneshjou. Large language models propagate race-based medicine. *npj Digital Medicine*, 6(1): 195, 2023. doi: 10.1038/s41746-023-00939-z. URL https://www.nature.com/articles/s41746-023-00939-z.

[25] Charles Nimo, Tobi Olatunji, Abraham Toluwase Owodunni, Tassallah Abdullahi, Emmanuel Ayodele, Mardhiyah Sanni, Ezinwanne C. Aka, Folafunmi Omofoye, Foutse Yuehgoh, Timothy Faniran, Bonaventure F. P. Dossou, Moshood O. Yekini, Jonas Kemp, Katherine A. Heller, Jude Chidubem Omeke, Chidi Asuzu Md, Naome A. Etori, Aïmérou Ndiaye, Ifeoma Okoh, Evans Doe Ocansey, Wendy Kinara, Michael L. Best, Irfan Essa, Stephen Edward Moore, Chris Fourie, and Mercy Nyamewaa Asiedu. AfriMed-QA: A pan-African, multi-specialty, medical question-answering benchmark dataset. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1948–1973, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.96. URL https://aclanthology.org/2025.acl-long.96/. Best Social Impact Paper Award.

[26] Silas Ruhrberg Estévez, Nicolás Astorga, and Mihaela van der Schaar. Timely clinical diagnosis through active test selection. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2510.18988.

[27] Zhoujian Sun, Cheng Luo, Ziyi Liu, and Zhengxing Huang. Conversational disease diagnosis via external planner-controlled large language models. *arXiv preprint arXiv:2404.04292*, 2024. URL https://arxiv.org/abs/2404.04292.

[28] Zhiyuan Hu, Chumin Liu, Xidong Feng, Yilun Zhao, See-Kiong Ng, Anh Tuan Luu, Junxian He, Pang Wei Koh, and Bryan Hooi. Uncertainty of thoughts: Uncertainty-aware planning enhances information seeking in LLMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://openreview.net/forum?id=CVpuVe1N22.

[29] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

[30] Lars Klein, Nearchos Potamitis, Roland Aydin, Robert West, Caglar Gulcehre, and Akhil Arora. Fleet of agents: Coordinated problem solving with large language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. URL https://openreview.net/forum?id=wjEkZIZdQT.

[31] Junying Chen, Chi Gui, Anningzhe Gao, Ke Ji, Xidong Wang, Xiang Wan, and Benyou Wang. CoD, towards an interpretable medical agent using chain of diagnosis. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14345–14368, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.740. URL https://aclanthology.org/2025.findings-acl.740/.

[32] Arsene Fansi Tchango, Rishab Goel, Zhi Wen, Julien Martel, and Joumana Ghosn. DDXPlus: A new dataset for automatic medical diagnosis. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/cae73a974390c0edd95ae7aeae09139c-Abstract.html.

[33] Jeffrey C. Hoch, Kumaran Baskaran, Harrison Burr, John Chin, Hamid R. Eghbalnia, Toshimichi Fujiwara, Michael R. Gryk, Takeshi Iwata, Chojiro Kojima, Genji Kurisu, Dmitri Maziuk, Yohei Miyanoiri, Jonathan R. Wedell, Colin Wilburn, Hongyang Yao, and Masashi Yokochi. Biological magnetic resonance data bank. *Nucleic Acids Research*, 51(D1):D368–D376, 2023. doi: 10.1093/nar/gkac1050.

[34] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[35] Hei Chan and Adnan Darwiche. On the revision of probabilistic beliefs using uncertain evidence. *Artificial Intelligence*, 163(1):67–90, 2005. doi: 10.1016/j.artint.2004.09.005.

[36] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997.

[37] Luis Enrique Sucar. *Probabilistic Graphical Models: Principles and Applications*. Springer, 2015.

[38] Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 337–346, 2011.

[39] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. URL https://arxiv.org/abs/2207.05221.

[40] Samuel Schmidgall, Rojin Ziaei, Carl Harris, Ji Woong Kim, Eduardo Pontes Reis, Jeffrey Jopling, and Michael Moor. AgentClinic: A Multimodal Benchmark for Tool-Using Clinical AI Agents. *npj Digital Medicine*, 2026. doi: 10.1038/s41746-026-02674-7. URL https://www.nature.com/articles/s41746-026-02674-7.

[41] Mohammad Almansoori, Komal Kumar, and Hisham Cholakkal. MedAgentSim: Self-Evolving Multi-Agent Simulations for Realistic Clinical Interactions. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2025*, volume LNCS 15968, pages 362–372. Springer Nature Switzerland, September 2025. doi: 10.1007/978-3-032-05114-1_35. URL https://papers.miccai.org/miccai-2025/0537-Paper2575.html.

[42] Zhihao Fan, Lai Wei, Jialong Tang, Wei Chen, Siyuan Wang, Zhongyu Wei, and Fei Huang. AI hospital: Benchmarking large language models in a multi-agent medical interaction simulator. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10183–10213, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.680/.

[43] Daeun Kyung, Hyunseung Chung, Seongsu Bae, Jiho Kim, Jae Ho Sohn, Taerim Kim, Soo Kyung Kim, and Edward Choi. PatientSim: A persona-driven simulator for realistic doctor-patient interactions. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=1THAjdP4QJ. Spotlight.

[44] Yusheng Liao, Yutong Meng, Yuhao Wang, Hongcheng Liu, Yanfeng Wang, and Yu Wang. Automatic interactive evaluation for large language models with state aware patient simulator. *arXiv preprint arXiv:2403.08495*, 2024. URL https://arxiv.org/abs/2403.08495.

[45] Thomas Savage, Ashwin Nayak, Robert Gallo, Ekanath Rangan, and Jonathan H. Chen. Diagnostic reasoning prompts reveal the potential for large language model interpretability in medicine. *npj Digital Medicine*, 7(1):20, 2024. doi: 10.1038/s41746-024-01010-1. URL https://www.nature.com/articles/s41746-024-01010-1.

[46] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, pages 4878–4887, 2017. URL https://papers.neurips.cc/paper/7073-selective-classification-for-deep-neural-networks.

[47] Nearchos Potamitis, Lars Henning Klein, Bardia Mohammadi, Chongyang Xu, Attreyee Mukherjee, Niket Tandon, Laurent Bindschaedler, and Akhil Arora. Cache saver: A modular framework for efficient, affordable, and reproducible LLM inference. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 25703–25724, 2025.

[48] Christopher Chiu, Silviu Pitis, and Mihaela van der Schaar. Simulating viva voce examinations to evaluate clinical reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2510.10278.

[49] Xavier Theimer-Lienhard, Mushtaha El-Amin, Fay Elhassan, Sahaj Vaidya, Victor Cartier-Negadi, David Sasu, Lars Klein, and Mary-Anne Hartley. Fully open meditron: An auditable pipeline for clinical LLMs. *arXiv preprint arXiv:2605.16215*, 2026. URL https://arxiv.org/abs/2605.16215.

[50] Yusuf Kesmen, Aniruddha Mukherjee, Yena Chang, David Sasu, Trevor Brokowski, Alexandra V. Kulinkina, Kristina Keitel, Akhil Arora, Lars Henning Klein, and Mary-Anne Hartley. CLIMB: A clinical multimorbidity benchmark for diagnosing co-occurring conditions through multiturn conversations. *arXiv preprint arXiv:2609.35462*, 2026. URL https://arxiv.org/abs/2609.35462.

[51] Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. MEDITRON-70B: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*, 2023.

[52] Andrew Sellergren, Sahar Kazemzadeh, Tiam Jaroensri, Atilla Kiraly, Madeleine Traverse, Timo Kohlberger, Shawn Xu, Fayaz Jamil, Cián Hughes, Charles Lau, Justin Chen, Fereshteh Mahvar, Liron Yatziv, Tiffany Chen, Bram Sterling, et al. MedGemma technical report. *arXiv preprint arXiv:2507.05201*, 2025.

[53] Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. HuatuoGPT-o1, towards medical complex reasoning with LLMs. *arXiv preprint arXiv:2412.18925*, 2024.

[54] Yichun Feng, Jiawei Wang, Lu Zhou, Yikai Zheng, Zhen Lei, and Yixue Li. Real-world doctor agent with proactive consultation through multi-agent reinforcement learning. *arXiv preprint arXiv:2505.19630*, 2025.

[55] Kwan Ho Ryan Chan, Yuyan Ge, Edgar Dobriban, Hamed Hassani, and René Vidal. Conformal information pursuit for interactively guiding large language models. In *Advances in Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=xAHozxfuUW`.

[56] Qipeng Wang, Rui Sheng, Yafei Li, Huamin Qu, Yushi Sun, and Min Zhu. MedKGI: Iterative differential diagnosis with medical knowledge graphs and information-guided inquiring. *arXiv preprint arXiv:2512.24181*, 2025. URL `https://arxiv.org/abs/2512.24181`.

[57] Hui Min Wong, Philip Heesen, Pascal Janetzky, Martin Bendszus, and Stefan Feuerriegel. MedClarify: An information-seeking AI agent for medical diagnosis with case-specific follow-up questions. *arXiv preprint arXiv:2602.17308*, 2026. URL `https://arxiv.org/abs/2602.17308`.

Appendix overview

This supplementary material is organized as follows:

A. Theoretical foundations 17
A.1 Soft evidence 17
A.2 EIG derivation and question-selection ablations 17
A.3 Question budget analysis 18
A.4 Confidence threshold tuning 18
B. System details 19
B.1 MOBAYES diagnostic procedure 19
B.2 Prompt templates 19
B.3 Standalone doctor protocol 23
B.4 Knowledge base construction 24
B.5 Patient simulation 28
B.6 Implementation details 29
C. Extended experimental results 29
C.1 LLM-elicited knowledge base, full results 29
C.2 Prediction normalisation 29
C.3 Confidence threshold experiments 29
C.4 Question-selection policy ablations 30
C.5 Prior distribution sensitivity 31
C.6 Accuracy-coverage analysis 32
C.7 Computational cost analysis 33
C.8 Persona robustness 33
C.9 Accuracy by disease prevalence 34
C.10 Failure mode analysis 35
C.11 Disease scaling analysis 37
C.12 Diagnostic session walkthroughs 37

C.13 Bootstrap confidence intervals 40
C.14 Baseline protocols and adaptations 40
C.15 True diagnosis absent from the KB 42
C.16 Sensitivity to KB noise and coverage 42
C.17 Parser audit 43
C.18 Model-matched sensor and standalone pairings 43
C.19 Full AgentClinic-MedQA 43
C.20 Diagnosis under a staged examination protocol 43
D. Extended related work 44

A Theoretical foundations

A.1 Soft evidence

Standard Bayesian conditioning assumes evidence is observed with certainty. In conversational diagnosis, patient responses carry intrinsic uncertainty: "I think I had a fever" is not equivalent to a confirmed temperature reading. We model this with **Pearl-style virtual evidence** (34; 35), also called *soft evidence*. Concretely, we augment the model with a virtual indicator node V whose likelihood ratio encodes the parser's confidence $c \in (0, 1]$:

$$P(V \mid X_f{=}v) = 1, \qquad P(V \mid X_f{\neq}v) = 1 - c. \tag{2}$$

For unambiguous responses ($c{=}1$), the standard Bayesian update applies:

$$b_{t+1}(d) \;\propto\; b_t(d) \cdot P(X_f{=}v \mid d). \tag{3}$$

For uncertain responses, the effective likelihood mixes with a neutral vector:

$$L_{\text{eff}}(d) \;=\; c \cdot P(X_f{=}v \mid D{=}d) \;+\; (1 - c) \cdot 1, \qquad b_{t+1}(d) \;\propto\; b_t(d) \cdot L_{\text{eff}}(d). \tag{4}$$

At $c{=}1$, V deterministically asserts the observation and we recover standard Bayes. At $c{\to}0$, the likelihood ratio $P(V \mid X_f{=}v) : P(V \mid X_f{\neq}v) \to 1{:}1$, so V carries no information and the posterior is unchanged. This semantics matches the parser's role: c is the system's confidence in its own extraction, not a posterior probability over the underlying event.

Why ordinal labels? The five confidence labels (*very_likely* through *very_unlikely*) provide a coarse but principled discretisation of patient certainty. This avoids requiring the LLM to output calibrated continuous probabilities. The fixed mapping to numerical weights ($c \in \{1.0, 0.80, 0.50, 0.25, 0.05\}$) is auditable and eliminates a potential source of unchecked model influence on diagnostic outcomes. Table 3 lists the complete mapping.

Table 3: Soft-evidence confidence weights. A three-tier cascade assigns labels: (1) deterministic pattern matching for clear or hedged language, (2) LLM-based extraction for ambiguous responses, (3) heuristic downgrade to `uncertain` when uncertainty cues are detected.

Label	c	When assigned
`very_likely`	1.00	Clear, direct: "yes", "no", "definitely"
`likely`	0.80	Requires interpretation or inference
`uncertain`	0.50	"I think so", "maybe", "not sure"
`unlikely`	0.25	Tangential or partially related
`very_unlikely`	0.05	Near-zero signal; contradictory

A.2 EIG derivation and question-selection ablations

EIG computation details. The EIG policy (Eq. 7 in main text) selects the unasked feature that maximally reduces expected posterior entropy. The predictive distribution over feature values is:

$$P(X_f{=}v \mid \mathcal{E}_t) \;=\; \sum_{d=1}^{K} P(X_f{=}v \mid d)\; b_t(d) \tag{5}$$

For each candidate value $v \in \mathcal{V}_f$, a counterfactual posterior is computed via a standard Bayesian update at full confidence ($c{=}1$) without modifying internal state:

$$b_t^{f=v}(d) \;=\; \frac{b_t(d) \cdot P(X_f{=}v \mid d)}{\sum_{d'} b_t(d') \cdot P(X_f{=}v \mid d')} \tag{6}$$

The EIG is the reduction in expected entropy:

$$\text{EIG}(f) \;=\; H(\mathbf{b}_t) \;-\; \sum_{v \in \mathcal{V}_f} P(X_f{=}v \mid \mathcal{E}_t) \cdot H(\mathbf{b}_t^{f=v}) \tag{7}$$

where $H(\mathbf{b}) \;=\; -\sum_d b(d) \log_2 b(d)$. At each turn the system selects the unasked feature that maximises this quantity:

$$f^* \;=\; \arg\max_{f \in \mathcal{F} \setminus \mathcal{A}_t} \text{EIG}(f). \tag{8}$$

Computational cost. At each turn t, the engine evaluates EIG for every unasked feature. Let $N_t = |\mathcal{F} \setminus \mathcal{A}_t|$ denote the remaining features. For each feature f, the inner loop iterates over $|\mathcal{V}_f|$ hypothetical values, each requiring a K-dimensional posterior update. The per-turn cost is therefore:

$$\mathcal{O}\left(N_t \cdot \max_f |\mathcal{V}_f| \cdot K\right) \tag{9}$$

For DDXPlus (N=314 features, $\max_f |\mathcal{V}_f|$=10, K=49 diseases), this yields ∼154K operations per turn, computed in ∼50 ms on a single CPU core with vectorised NumPy. Since N_t decreases by one each turn, the total session cost is $O(N \cdot T \cdot \max_f |\mathcal{V}_f| \cdot K)$, dominated by the first few turns.

Top-k focused EIG. Once the posterior concentrates ($\max_d b_t(d) \geq \tau_{\text{focus}}$), an additional EIG term encourages fine-grained discrimination among the most probable hypotheses:

$$\text{EIG}_{\text{combined}}(f) \;=\; \text{EIG}_{\text{global}}(f) \;+\; \lambda \cdot \text{EIG}_{\text{top-}k}(f) \tag{10}$$

where $\text{EIG}_{\text{top-}k}$ is computed over only the top-k diseases (renormalised). Ablation results (Table 10) show that focused top-k EIG with k=3, λ=0.5, τ_{focus}=0.3 improves top-1 accuracy by +12 pp over the global baseline. The main experiments use global EIG (λ=0) to keep the default policy parameter-free; focused EIG is available as an optional enhancement.

A.3 Question budget analysis

Ideal lower bound. Under a uniform prior over K diseases, the initial entropy is $H_0 = \log_2 K$ bits. A perfectly informative binary question removes at most 1 bit, so at least $\lceil \log_2 K \rceil$ questions are required in the noiseless case.

Effect of soft evidence. In practice, evidence is soft: the effective likelihood $L_{\text{eff}}(d) = c{\cdot}P(X_f{=}v \mid d) + (1{-}c)$ (Eq. 4) corresponds to Pearl-style virtual evidence with a virtual node V_c defined by $P(V_c \mid X_f{=}v) = 1$ and $P(V_c \mid X_f{\neq}v) = 1 - c$ (Appendix A.1). Since V_c is generated by passing X_f through a c-parameterised stochastic channel, $D \to X_f \to V_c$ forms a Markov chain. By the data processing inequality:

$$I(D;\, V_c) \;\leq\; I(D;\, X_f), \tag{11}$$

i.e., the information about the disease accessible to the engine via the soft observation V_c is upper-bounded by the information available in the clean observation X_f. The reduction is monotonic in c: at c=1, V_c deterministically reveals X_f and we recover the full $I(D; X_f)$; at c=0, V_c is independent of X_f and carries no information. The empirical average confidence $\bar{c} \approx 0.9$ on our benchmarks means the information loss per question is modest.

Budget and stopping-rule parameters. The warm-up $T_{\min}$ suppresses early-stopping during the first few turns, when posteriors are dominated by the chief complaint and prior spread is largest; we use the same $T_{\min}$ for every backend so that differences in turn budget across rows of Table 2 reflect the policy and not a hand-tuned schedule. Even under adversarial patient personas (Appendix C.8), where noisy or withheld responses reduce per-turn information, the turn budget suffices for convergence.

A.4 Confidence threshold tuning

The confidence threshold τ controls the trade-off between coverage (fraction of patients who receive a diagnosis) and selective accuracy (quality among diagnosed patients).

DDXPlus. We sweep $\tau \in \{0.00, 0.05, \ldots, 0.95\}$ over the posterior confidence $\max_d b_T(d)$. Patients with $\max_d b_T(d) < \tau$ are abstained. τ^* is tuned by leave-one-out on a 50-case set. Results are reported in Appendix C.3.

LLM-generated KBs. The same procedure applies. Because the candidate set is smaller and the posterior tends to be less peaked, the DHS-optimal τ^* is lower than on DDXPlus. The full sweep is reported in Appendix C.3.

Global vs. per-disease calibration. We use a global threshold (one τ for all diseases) rather than per-disease thresholds τ_d because the limited set size (n=50) yields too few positive examples per disease for reliable per-disease calibration. The full threshold sweep is reported in Appendix C.3.

B System details

B.1 MOBAYES diagnostic procedure

Algorithm 1 formalises the diagnostic loop described in Section 3: bulk-extracted chief-complaint evidence is folded into the prior, then each subsequent turn selects the most informative feature via expected information gain (EIG), updates the posterior under Pearl-style soft evidence, and checks the warm-up + threshold stopping rule. Symbols match those introduced in Sections 3.1–3.4.

Algorithm 1 MOBAYES Diagnostic Session

Require: Knowledge base $\mathcal{K}$ with prior $\boldsymbol{\pi}$, likelihoods $\{P(X_f{=}v \mid d)\}$, schema; threshold τ; warm-up $T_{\min}$; budget $T_{\max}$
Ensure: Diagnosis d^* or ABSTAIN

Initialise $b_0(d) \leftarrow \pi(d)$ for all $d \in \mathcal{D}$
LLM bulk-extracts $\{(f_j, v_j, c_j)\}_{j=1}^{J}$ from chief complaint ▷ §3.4
for $j = 1, \ldots, J$ **do**
    $\mathbf{b} \leftarrow \textsc{UpdateBelief}(\mathbf{b}, f_j, v_j, c_j)$ ▷ Eq. (4)
end for
for $t = 1, \ldots, T_{\max}$ **do**
    $f^* \leftarrow \arg\max_{f \notin \mathcal{A}_t} \mathrm{EIG}(f;\, \mathbf{b}_t)$ ▷ Eq. (7)
    LLM verbalises f^* as natural-language question
    $(v_t,\, c_t) \leftarrow \textsc{Parse}(\text{patient response}, f^*, \mathcal{S}_{f^*})$
    if $v_t =$ `unknown` or $v_t =$ `clarification` **then**
        Re-ask f^* with clarification; parse again $\rightarrow (v_t, c_t)$
    end if
    $\mathbf{b}_{t+1} \leftarrow \textsc{UpdateBelief}(\mathbf{b}_t, f^*, v_t, c_t)$
    if $t \geq T_{\min}$ **and** $\max_d b_{t+1}(d) \geq \tau$ **then**
        break ▷ sufficient evidence gathered
    end if
end for

Decision:
$d^* \leftarrow \arg\max_d b_T(d)$
if $b_T(d^*) \geq \tau$ **then return** d^*
else return ABSTAIN
end if

Paradigm comparison. Figure 4 contrasts the two paradigms at a glance: existing state-of-the-art systems emulate abductive reasoning inside the LLM (hidden chain-of-thought, multi-agent role-play, pseudo-probability prompts, planner / tool use), producing free-form answers without calibrated uncertainty, accountability, or audit trail. MOBAYES separates language and reasoning: the LLM only parses patient utterances and verbalises selected questions, while a Bayesian module maintains an explicit belief state, updates it with soft evidence, selects the next question by expected information gain, and applies a calibrated stopping rule, all over a transparent and replaceable knowledge base.

B.2 Prompt templates

All prompts use temperature 0.0 for deterministic output. The LLM never receives posterior values, entropy, or any internal engine state.

Parsing prompt. The parsing prompt receives the patient utterance, the feature under inquiry, and the allowed schema values. A deterministic cascade (pattern matching for binary yes/no, emphatic affirmations, numeric ranges, and direct schema substring matches) is applied *before* the LLM call; the LLM is invoked only as a fallback for ambiguous utterances. The prompt enforces 13 confidence-assignment rules (e.g., prefer `unknown` over hard negatives when the response is partial or vague) to ensure conservative, faithful extraction.

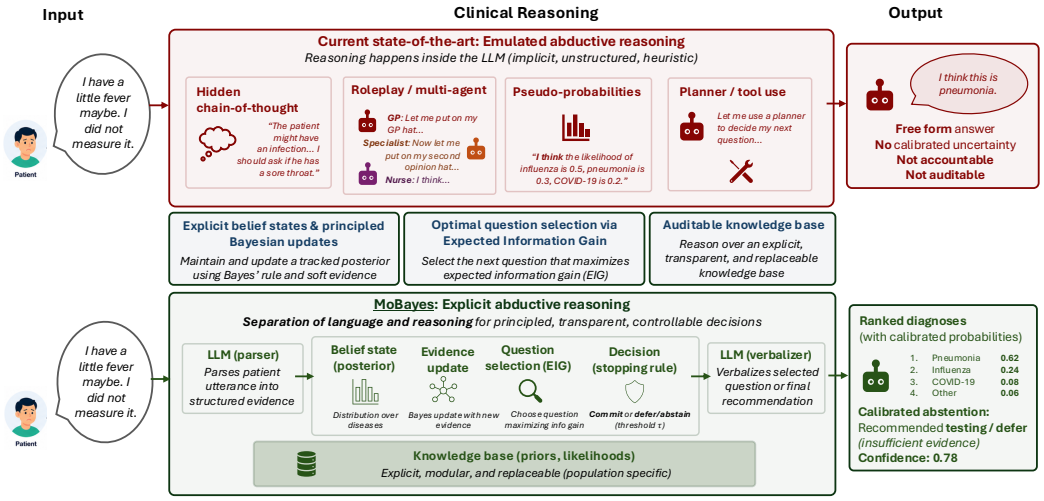


Figure 4: Paradigm comparison. **Top:** current state-of-the-art emulates abductive reasoning inside the LLM (hidden chain-of-thought, role-play, pseudo-probabilities, planner / tool use); the output is a free-form answer with no calibrated uncertainty, accountability, or audit trail. **Bottom:** MoBayes separates language and reasoning. The LLM acts only as a parser and verbaliser; an explicit Bayesian module tracks a posterior, applies soft-evidence updates, selects the next question by expected information gain, and decides whether to commit or abstain at threshold τ. Reasoning is conducted over a transparent, modular, replaceable knowledge base, and the output is a ranked differential with calibrated probabilities and a defensible abstain option.

Parsing Prompt

```
System: You are a medical data extractor.
User Response: "{utterance}"
Feature to Extract: "{feature}" (Allowed: {schema})

TASK:
1. Classify the user response into one of the Allowed Values.
2. Assess confidence: very_likely, likely, uncertain, unlikely, very_unlikely

KEY RULES:
- Direct answer ("Yes", "Not really"): map to closest value.
- Unrelated response: return "unknown|likely".
- Uncertain language ("I think so", "maybe"): use "uncertain".
- Prefer "unknown" over hard negative when partial/vague.

Return format: "value|confidence_level"
Examples: "high|likely", "yes|uncertain", "no|very_likely"
```

Verbaliser prompt. The verbaliser receives the feature name, answer schema, and a coarse system-confidence indicator derived from $\max_d b_t(d)$. It modulates conversational tone only; no internal state is exposed.

Verbaliser Prompt

```
System: You are a professional medical diagnostic assistant.
Task: Ask the patient a question to check for: {feature}.

CONTEXT:
- Feature Name: {feature}
- Expected Details: {schema}
- System Confidence: {low|medium|high}

CONSTRAINTS:
1. NEVER use technical IDs (e.g., "f_fever", "d_flu").
2. Speak naturally and empathetically.
```

```
3. Do NOT mention probabilities or internal values.
4. If clarifying a previous confusion, keep it brief.
```

Bulk intake prompt. At session start, a single bulk intake call maps the patient's opening narrative to multiple (f, v, c) triples simultaneously, reducing the number of follow-up questions needed.

Bulk Intake Prompt

```
System: You are an expert medical intake specialist.
User Text: "{narrative}"

TASK: Extract any clinical features mentioned in the text.

ALLOWED FEATURES:
- {feature_id}: {name} (Values: {values})

RULES:
1. Only extract explicitly mentioned or strongly implied features.
2. Extract demographics (age, gender, location) if present.
3. Do NOT infer negatives from silence; omit unlisted features.
4. Assess confidence for each extracted feature.

Return JSON: {"feature_id": {"value": "...", "confidence": "likely"},
"demographics": {"age": N, ...}}
```

Patient simulator prompt. The patient simulator receives the full clinical profile (demographics, chief complaint, symptoms, medical history, observed findings) and persona instructions. Crucially, the patient must answer faithfully according to its observed findings, including clear denials when the finding is negative, but must say "I'm not sure" for features not listed in its profile, avoiding fabrication.

Patient Simulator Prompt

```
System: You are a patient in an Emergency Department consultation.

PATIENT PROFILE:
- Age, Gender, Chief Complaint, Symptoms Present
- Social History, Family History, Medical History, Pain Level
- KNOWN OBSERVED FINDINGS (answer faithfully; do not assume unlisted findings)

RULES:
1. Answer based on KNOWN OBSERVED FINDINGS and patient profile.
2. If asked about something listed: answer faithfully (including denials).
3. If NOT listed: say ‘‘I’m not sure’’ or ‘‘I don’t know’’. Do not invent
symptoms.
4. NEVER reveal your diagnosis directly.
5. Keep responses concise (1-3 sentences).

PERSONA:
- Language: {CEFR level A/B/C}
- Personality: {plain | verbose | overanxious | distrustful}
- Memory: {high | low recall}
- Alertness: {normal | moderate daze | high daze}
```

Standalone doctor prompt. The standalone LLM doctor receives no external reasoning support. It conducts the full diagnostic interview and outputs a differential ([DDX]) or abstains ([ABSTAIN]).

Standalone Doctor Prompt , Single-Disease (DDXPlus)

```
System: You are an emergency department physician conducting a diagnostic
interview.
Ask focused, one-at-a-time questions to narrow down the diagnosis.
After gathering enough information:
- Start with [DDX] followed by your top 5 diagnoses (most to least likely),
separated by commas.
- If not confident, start with [ABSTAIN] + short reason, then [POSSIBLE] + your
top 3 possible diagnoses.
```

```
Guidelines:
- Ask one question at a time.
- Focus on discriminating features.
- Consider common and serious diagnoses.
- Do not guess when evidence is insufficient.
- Maximum of {max_turns} questions before [DDX]/[ABSTAIN].
```

Disease matching prompt. Free-text doctor outputs from every non-AC baseline are aligned to the active closed disease list by a unified LLM-judge pass with the gpt-5.4-nano backbone shared across all baselines. A single call returns the closed-list Top-1, the closed-list Top-3, and an explicit *abstention* flag (true only on a literal refusal-to-commit). Off-list commits are recorded as committed-but-wrong, not as abstentions; this separation is what lets selective accuracy and coverage decouple in Table 21.

Disease Matching Prompt

```
You are a medical-coding assistant.  Given the doctor's diagnostic output below,
do three things:

1.  TOP-1:  pick the disease from the closed list of {n} that semantically
matches the doctor's TOP candidate (the rank-1 commit).  If none match, output
"Unknown".

2.  TOP-3:  pick up to three diseases from the closed list that semantically
match the doctor's RANKED differential (in order).  If a candidate cannot be
mapped, output "Unknown" for that slot.

3.  ABSTAIN: set "abstained":  true only if the doctor's TOP candidate text
is literally ABSTAIN, Unknown, I don't know, empty, or otherwise an explicit
refusal.  Set false whenever the doctor names any disease, even if that disease
is not in the closed list (off-list = committed-but-wrong, not abstain).

Closed disease list:
{disease_list}

Doctor's diagnostic output:
--
{diag_text}
--

Semantic-match guidance:
- ''Chronic Obstructive Pulmonary Disease (COPD)'' matches ''Chronic obstructive
pulmonary disease''.
- ''CKD'' matches ''Chronic kidney failure''.
- ''Acute acalculous cholecystitis with sepsis'' matches ''Acalculous
cholecystitis''.
- Markdown asterisks, parenthetical abbreviations, and minor wording variants are
all OK.
- Use "Unknown" only if the candidate clearly does not refer to any listed
disease.

Output JSON exactly in this format and nothing else:
{"top1":  "<disease|Unknown>", "top3":  ["<d1>", "<d2>", "<d3>"], "abstained":
true|false}
```

LLM-generated KB prompts. The LLM-generated KB is constructed via two sequential prompts. The *feature generation prompt* asks the model to propose clinically plausible features for a given disease (with deduplication against an existing catalog). The *distribution estimation prompt* then elicits $P(\text{feature} \mid \text{disease})$ for all feature–disease pairs: binary features as a single `prob_yes`, and continuous features as a discrete distribution over a 0–5 severity scale (where 0 encodes absence). Features are processed in chunks; outputs are validated (distributions must sum to 1) before being added to the KB.

Feature Generation Prompt

```
You are helping build a synthetic feature catalog for a conversational diagnostic
system.
Disease:  {disease_id}, {name}
```

```
Include only features with P(present | disease) > {threshold}.
Allowed kinds: symptom_presence, symptom_attribute, medical_history, lifestyle,
risk_factor.
Before proposing, check the existing catalog and reuse semantically equivalent
features.
Return: action=reuse|new, estimated_probability, justification.
```

Distribution Estimation Prompt

```
You are generating likelihood parameters for a Bayesian diagnostic model.
Disease: {disease_id}, {name}
For binary features: return prob_yes.
For numerical features (0-5 scale, 0=absent): return probability for each value;
must sum to 1.
Use medically plausible values; avoid extreme certainty.
```

AgentClinic-MedQA KB construction prompts. The AgentClinic-MedQA KB (Section B.4) is built in a multi-stage pipeline rather than a single elicitation pass. Disease-driven and patient-driven feature elicitation prompts follow the same template as the LLM-generated KB feature prompt above (with the closed disease list and patient narratives respectively as context). The three remaining prompts encode the constraints specific to this column: a deduplication pass that preserves both generic and specific feature variants, and the banded continuous likelihood elicitation.

Preserve-Specificity Dedup Prompt

```
You are a clinical knowledge engineer building a diagnostic feature schema. You
will receive candidate phrases from two source catalogs:
  - ''specific'': disease-driven detailed phrasings (e.g., ''rough hyperkeratotic
plaque on sun-exposed area'').
  - ''generic'': patient-driven short phrasings (e.g., ''skin lesions'').
Many are near-duplicates. Produce a canonical list that PRESERVES BOTH LAYERS.
Rules:
  (1) DROP entirely: negative phrasings, demographic-only items, lab / imaging
results, metadata-only items.
  (2) KEEP both general and specific variants of the same observation; DO NOT
merge across granularity.
  (3) MERGE only true duplicates (same concept, same specificity level).
  (4) Use clinical phrasing, lowercase, no patient slang.
Output a JSON array of canonical feature names, ONLY the array.
```

Banded Likelihood Elicitation Prompt

```
You are a clinical knowledge engineer assigning P(YES | disease) likelihoods.
Disease: {disease}.
For each relevant feature, assign a probability on a CONTINUOUS scale, using
the following bands as a guide (vary values WITHIN bands; do NOT cluster on band
midpoints):
  - 0.02 default (background; unrelated organ system).
  - 0.05-0.10 slight relevance (rare incidental association).
  - 0.12-0.25 possible / uncommon presentation.
  - 0.40-0.60 common presentation.
  - 0.65-0.85 strong / typical / classic presentation.
  - 0.90-0.97 pathognomonic / near-defining.
For features irrelevant to this disease, OMIT them; they will fall through to the
0.02 default.
Output JSON: {"disease": "...", "p_yes": {"feature_name": value, ...}}.
```

B.3 Standalone doctor protocol

Algorithm 2 summarises the standalone doctor inference loop.

The standalone doctor outputs a comma-separated differential ([DDX]) or explicitly abstains ([ABSTAIN]). The prompt requests the top 5 diagnoses, ranked by likelihood. Crucially, the doctor generates free-text disease names *without access to the KB disease list*, mirroring realistic deployment

Algorithm 2 Standalone LLM Doctor

Require: Patient profile $\mathbf{p}$, LLM doctor $\mathcal{M}$, turn budget $T_{\max}$
Ensure: Diagnosis $\hat{y}$ or ABSTAIN

$\mathcal{H} \leftarrow [\,]$ ▷ dialogue history
$\mathcal{H}$.append(SystemPrompt($\mathcal{M}, T_{\max}$))
$r_{\text{patient}} \leftarrow$ ChiefComplaint($\mathbf{p}$)
for $t = 1, \ldots, T_{\max}$ **do**
  $\mathcal{H}$.append(r_{patient})
  $r_{\text{doctor}} \leftarrow \mathcal{M}(\mathcal{H})$
  $\mathcal{H}$.append(r_{doctor})
  if r_{doctor} contains `[DDX]` **then**
    Parse comma-separated list $\rightarrow \hat{\mathbf{y}}_{\text{raw}} = [\hat{y}_1, \ldots, \hat{y}_k]$
    $\hat{\mathbf{y}} \leftarrow$ MatchToKB($\hat{\mathbf{y}}_{\text{raw}}, \mathcal{D}_{\text{KB}}$) ▷ LLM-based name matching
    return $\hat{\mathbf{y}}$
  else if r_{doctor} contains `[ABSTAIN]` **then**
    return ABSTAIN (+ optional `[POSSIBLE]` list)
  end if
  $r_{\text{patient}} \leftarrow$ PatientSim($\mathbf{p}, \mathcal{H}$)
end for
Inject "provide your final diagnosis now" into $\mathcal{H}$
$r_{\text{doctor}} \leftarrow \mathcal{M}(\mathcal{H})$
if r_{doctor} contains `[DDX]` **then**
  Parse and $\hat{\mathbf{y}} \leftarrow$ MatchToKB($\hat{\mathbf{y}}_{\text{raw}}, \mathcal{D}_{\text{KB}}$)
  return $\hat{\mathbf{y}}$
else return ABSTAIN
end if

where a physician names conditions from memory. We additionally evaluate a closed-world variant (CW) where the doctor receives the KB disease list at inference time in Section 5; the CW baseline directly addresses whether KB-list awareness changes standalone performance.

Post-hoc disease name normalisation. Free-text predictions (e.g., "Acute exacerbation of COPD") may differ lexically from KB entries (e.g., "Acute COPD exacerbation / infection"), so the MatchToKB step of Algorithm 2 aligns each prediction to the canonical disease list using the unified LLM-judge described in Appendix B.2. Predictions that the judge cannot match to any closed-list entry are retained as off-list commits and contribute neither to Top-k accuracy nor to abstention, ensuring that naming a disease outside the KB scope incurs a penalty. We manually audited all 250 standalone-doctor commits across the six tested models on DDxPlus50 (sessions where the doctor produced a final `[DDX]`; abstention sessions do not invoke the matcher). Between 4 and 13 (1.6% to 5.2%, 95% Clopper-Pearson CI [0.4%, 8.7%]) involved a matcher error that mapped a clinically distinct entity onto the ground-truth KB entry, generating a false-positive top-1 hit. The range reflects the strictness of clinical equivalence (e.g., "acute decompensated heart failure" vs "acute pulmonary edema"). No false negatives were observed. Matcher errors thus inflate standalone scores; correcting them would strengthen rather than weaken MOBAYES's reported advantage.

The matcher (one GPT-5.4-nano call per session) adds approximately 0.018 cents per standalone session: under 0.1% of the cost of the most expensive standalone doctors and at most 2% of the cheapest. It is omitted from Table 13 without affecting any reported value at the displayed precision.

B.4 Knowledge base construction

Both knowledge bases are clinically grounded, each through a different mechanism: DDXPlus derives likelihoods from physician-designed patient records, and the LLM-generated KB from structured elicitation of a frontier model's medical knowledge. Table 4 provides a summary comparison.

Both KBs share the same Dirichlet–Categorical likelihood formulation. For each disease d and feature f with values $\mathcal{V}_f$, we store empirical counts $n_{f,v,d}$ (co-occurrence in training records, or normalised pseudo-counts derived from LLM elicitations) and define the conditional as a Categorical

Table 4: Summary statistics of the knowledge bases used in evaluation.

	DDXPlus	**LLM-KB (GPT)**	**LLM-KB (Gemini)**
Source	Empirical	GPT-5.4	Gemini 3.1
Diseases	49	18	18
Features	314	160	89
Binary	306	154	85
Categorical / ordinal	8	6	4
KB source records	∼1.03M	–	–
Evaluation patients	50	50	50

with additive Laplace smoothing:

$$P(X_f{=}v \mid D{=}d) \;=\; \frac{\alpha_{f,v,d}}{\sum_{v' \in \mathcal{V}_f} \alpha_{f,v',d}}, \qquad \alpha_{f,v,d} = n_{f,v,d} + 1. \tag{12}$$

The Laplace prior ensures $P(X_f{=}v \mid d) > 0$ for every triple, preventing zero-likelihood pathologies under sequential inference; counts are normalised to sum to 100 per (disease, feature) pair before smoothing.

DDXPlus-derived KB. DDXPlus (32) contains ∼1.03M training records, each annotated with a ground-truth pathology and a set of observed evidences. We map the 49 pathologies to canonical disease IDs with ICD-10 codes and organise them into 9 chapter-level categories. The 223 original evidences comprise three types: binary, categorical (V-code resolved to readable labels), and multi-choice. Multi-choice evidences are expanded into individual binary sub-features (top-20 values by frequency), yielding 314 features in total.

Co-occurrence counts are accumulated per (disease, feature, value) triple across all training patients. Binary evidences absent from a patient's record are treated as negative, except for a small set of negated features where this default would invert the clinical meaning. Priors are set proportional to empirical disease frequency in the training split.

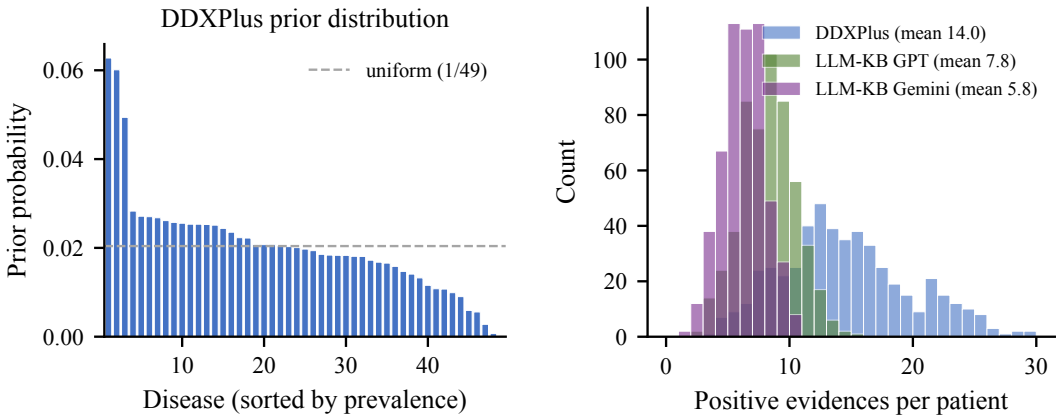


Figure 5: **Left:** DDXPlus prior distribution sorted by prevalence. The long tail ($\max / \min \approx 200\times$) reflects real-world disease frequency; the dashed line shows the uniform baseline. **Right:** Distribution of positive evidence count per evaluation patient across KBs.

LLM-generated KB. To control for informational advantages (Section 4.1), we construct LLM-generated KBs whose likelihoods are elicited entirely from frontier language models. We generate two variants: one using GPT-5.4 and one using Gemini 3.1, enabling a cross-model comparison of zero-shot medical knowledge.

Construction proceeds in two stages (prompts in Appendix B.2). In the *feature proposal* stage, the model generates clinically relevant features for each disease. Proposals reference a growing shared catalog to encourage feature reuse across diseases; a deduplication step removes near-duplicate entries. Features with estimated relevance below a threshold θ are discarded. In the *distribution estimation* stage, the model provides $P(\text{feature} \mid \text{disease})$ for all retained features: a single probability for binary

features, and a discrete distribution over a severity scale for ordinal features. All outputs undergo automated validation (probability constraints, identifier consistency) before inclusion; invalid entries are excluded. The resulting pseudo-counts are normalised identically to the DDXPlus KB.

AgentClinic-MedQA KB construction. AgentClinic-MedQA ships no labelled training split, so the frequency-based likelihood estimation used for DDXPlus is unavailable, and a single-pass elicitation from a fixed verbal scale induces a mode-collapse artefact in which most likelihoods cluster on a few discrete values, leaving the engine unable to discriminate symptomatically similar diseases. We therefore assemble the KB in two stages, keeping the benchmark's closed disease universe and its native moderator unchanged and setting priors to uniform.

The feature schema is the union of two complementary symptom catalogs. A *patient-driven* pass extracts findings from the validation-set patient narratives shipped with the benchmark (e.g. *skin lesions*), grounding the schema in the language patients actually use. A *disease-driven* pass elicits clinically discriminative findings from the authoring language model one disease at a time (e.g. *rough hyperkeratotic plaque on sun-exposed skin*), supplying the specific terminology a clinician would seek. An LLM deduplication pass then merges the two catalogs under a preserve-specificity constraint, so coarse and fine-grained variants of the same observation are retained side by side. The conditional likelihoods $P(\text{yes} \mid d)$ are subsequently authored by a language model under a continuous banded prompt that spreads probability mass across the full $[0, 1]$ interval, avoiding the bin-clustering artefact of the simpler verbal-scale elicitation. The benchmark's doctor, patient, measurement, and moderator agents are not modified; per-case adaptations are protocol-level only (Section C.14).

Synthetic patient generation. Since no real patient data exists for LLM-generated KBs, evaluation patients are generated through a parallel per-disease elicitation: for each disease d, the LLM is queried for the conditional probability $P(X_f \mid D{=}d)$ of each relevant clinical feature, and a feature vector is then drawn by an independent coin flip per feature,

$$x_f^{(i)} \sim P(X_f \mid D{=}d) \quad \text{for each feature } f \in \mathcal{F}_d, \tag{13}$$

a Bernoulli with parameter $P(X_f{=}\texttt{yes} \mid d)$ for binary features and a categorical draw over the 0–5 severity scale for ordinal ones, where $\mathcal{F}_d \subseteq \mathcal{F}$ is the set of features relevant to disease d (those with non-uniform likelihood). The resulting patients are independent samples from the LLM's own per-disease symptom distributions, not handcrafted vignettes. The chief complaint is constructed from the first three positive symptoms; the full feature map is stored as the patient's observed findings. Demographics (age, sex) are sampled uniformly at random. This yields $K \times n_{\text{per}}$ patients (e.g., $18 \times 30 = 540$), from which a stratified subset of $n{=}50$ is selected for benchmarking (at least one patient per disease). The sampling procedure ensures a "fair fight": both MOBAYES and standalone doctors face patients whose symptom profiles are consistent with the LLM's own clinical beliefs, neither system has an informational advantage.

Cross-KB transfer analysis. To test KB generalisation, we evaluate each ecosystem's MOBAYES on patients generated from the *other* KB. Let $\mathcal{F}^{\text{GPT}}$ and $\mathcal{F}^{\text{Gem}}$ denote the feature sets of the two KBs. Features are matched by canonical name: $\mathcal{F}^{\cap} = \{f : \text{name}(f) \in \text{names}(\mathcal{F}^{\text{GPT}}) \cap \text{names}(\mathcal{F}^{\text{Gem}})\}$. Of 160 GPT and 89 Gemini features, $|\mathcal{F}^{\cap}| = 45$ share identical names and types (28% of GPT, 51% of Gemini). When a cross-KB patient presents a feature $f \notin \mathcal{F}_{\text{KB}}$, the engine receives no matching schema entry and skips the update (equivalent to $L_{\text{eff}} = \mathbf{1}$). On average, GPT patients retain 43% of their features when evaluated against the Gemini KB, and Gemini patients retain 59% against the GPT KB. Results are reported in Table 5.

Table 5: Cross-KB transfer: each MOBAYES variant evaluated on patients from the other ecosystem. "Native" rows repeat the in-ecosystem results for reference. Feature coverage indicates the fraction of patient features recognised by the cross-KB.

System	Patients	Feat. Cov.	Top-1	Top-3	Sel. Acc	Cov.	DHS
MOBAYES + GPT-nano, KB: GPT	GPT (native)	100%	70	84	78	90	83
MOBAYES + GPT-nano, KB: GPT	Gemini (cross)	43%	52	78	54	96	69
MOBAYES + Gemini FL, KB: Gemini	Gemini (native)	100%	64	84	66	94	78
MOBAYES + Gemini FL, KB: Gemini	GPT (cross)	59%	42	66	48	84	61

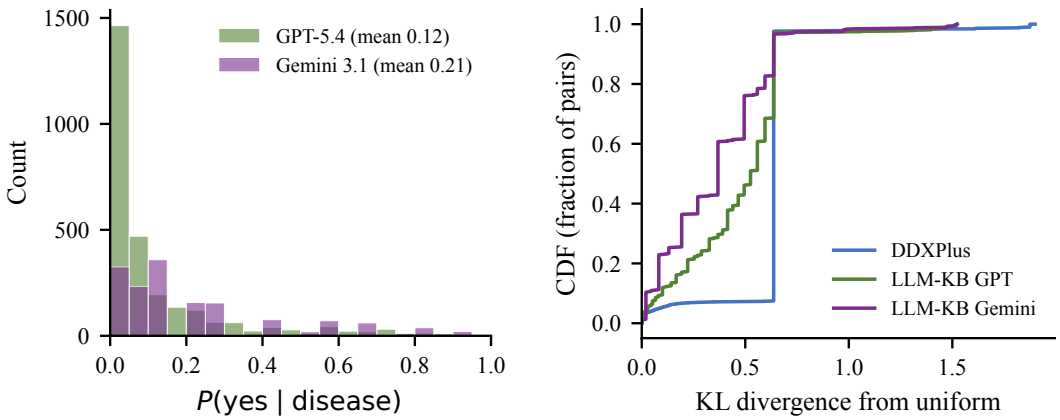


Figure 6: **Left:** Distribution of LLM-elicited binary likelihoods $P(\text{yes} \mid d)$ for both GPT and Gemini KBs; the strong left skew indicates that most disease–feature associations are weak. **Right:** CDF of per-pair KL divergence from uniform across all three KBs; DDXPlus (empirical) has the highest informativeness, while both LLM-KBs are comparable despite being synthetically generated. The informativeness gap between the empirical and LLM-elicited KBs is likely prompt-addressable: contrastive or pathognomonic-emphasis elicitation prompts that explicitly ask the model to discriminate symptomatically adjacent diseases should pull the LLM-KB CDF closer to the empirical curve.

Inter-model agreement on LLM-generated KBs. We compare the two LLM-generated KBs (GPT-5.4 and Gemini 3.1, same 18 diseases, same generation pipeline) to quantify inter-model agreement on medical knowledge. GPT generated 160 binary features; Gemini produced 89. Only 45 share identical names (Jaccard J=0.22), yet downstream agreement on likelihoods is high: on the 45 shared features across 18 diseases (810 pairs; Figure 7), Pearson r=0.931 with MAE=0.071. A systematic bias exists: Gemini assigns likelihoods 0.055 higher on average (Figure 7). Gemini's features are 1.7× more discriminative (cross-disease variance 0.041 vs 0.024; Figure 8), consistent with a smaller but more targeted feature set. Despite no shared weights, the two KBs converge on what matters, suggesting that frontier LLMs encode broadly consistent medical knowledge when prompted with structured elicitation.

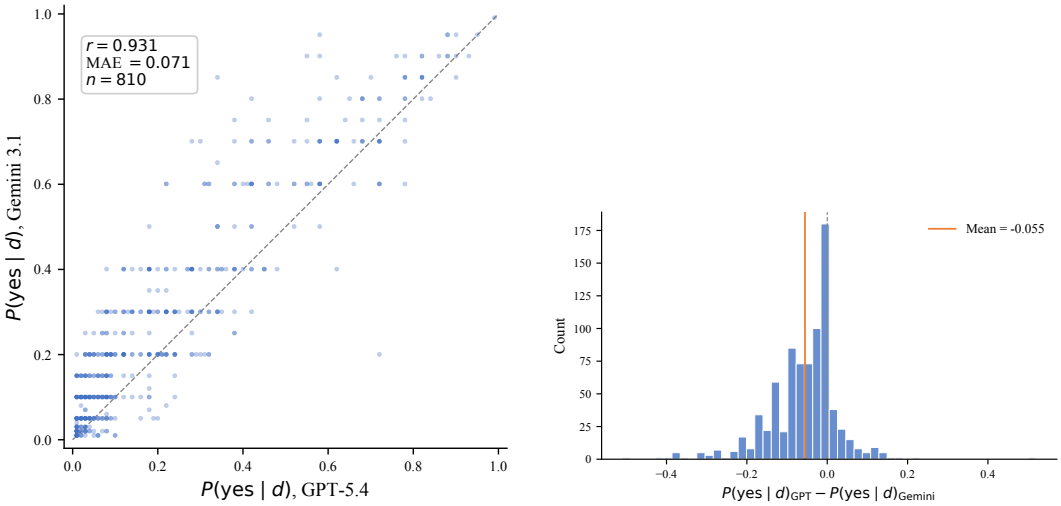


Figure 7: **Left:** Scatter plot of $P(\text{yes} \mid d)$ for 45 shared features across 18 diseases (n=810 pairs); dashed line is perfect agreement. **Right:** Distribution of pairwise likelihood differences; the left-skewed distribution (mean $= -0.055$) confirms Gemini's systematically higher assignments.

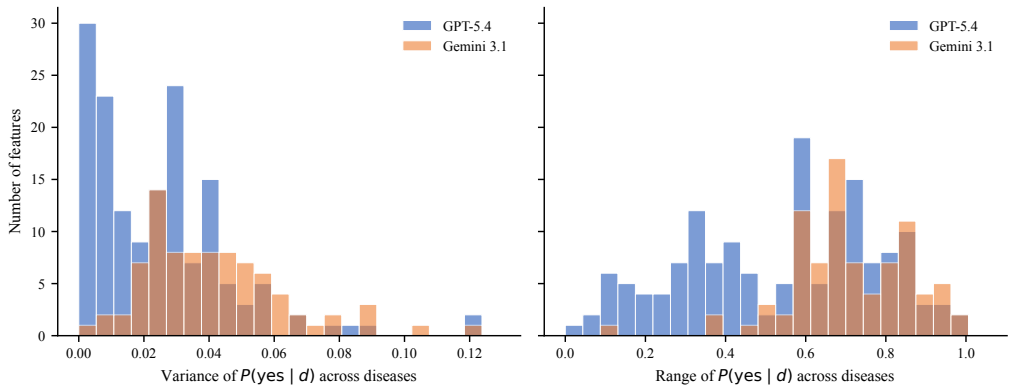


Figure 8: Feature discriminativeness (left: cross-disease variance; right: cross-disease range) for GPT-5.4 (154 features) and Gemini 3.1 (85 features). Gemini's features are more discriminative on average, reflecting both targeted feature selection and more extreme likelihood assignments.

B.5 Patient simulation

We simulate patients using an LLM-based conversational agent that receives a ground-truth clinical profile (demographics, chief complaint, symptom findings) and responds to doctor questions in character. The simulator is inspired by the persona taxonomy of PatientSim (43), which parameterises patient behaviour along four axes: *personality archetype* (cooperative, anxious, distrustful, etc.), *language proficiency*, *recall accuracy* (high/low), and *cognitive confusion* (none/moderate/high).

We select five personas by varying these axes (full definitions in Table 14): *Plain* (cooperative baseline), *Overanxious* (high anxiety, over-reports), *Distrustful* (withholds information), *Dazed* (low recall, moderate confusion), and *Verbose* (tangential, low recall).

The simulator LLM (GPT-5.4-nano) is shared across all experiments to ensure consistent patient behaviour. It sees only the clinical profile and persona instructions; it never accesses the diagnostic engine's internal state, posterior, or knowledge base. The simulator is instructed to answer faithfully for features listed in its profile and to express uncertainty for unlisted features, though we observe that LLMs occasionally infer plausible but ungrounded responses from profile context (see Appendix C.10 for the impact on diagnostic accuracy).

Patient-simulator adaptations across the three columns. Each external simulator is integrated through a thin compatibility shim that leaves its diagnostic behaviour, system prompt, and per-turn protocol unchanged; the changes are interface-level only.

(1) MEDDxAgent HT-sim. MEDDxAgent's history-taking simulator is plugged in as released; the patient system prompt, including the upstream instruction to default to "I don't know" for any feature outside the profile, is unchanged.

(2) AgentClinic native patient agent. AC's PatientAgent is used as released, with its system prompt and turn-by-turn conversation accumulator preserved verbatim. The lenient parsing prompt of Appendix B.2 is swapped in on the *engine* parser to handle hedged AC patient utterances; the patient simulator itself is untouched. Patient- and doctor-side token counts are tracked separately on this column.

(3) PatientSim on DDxPlus. We adopt the four-axis persona taxonomy of PatientSim (43) (personality, language proficiency, recall accuracy, cognitive confusion) along with the upstream persona category names. The reference implementation is built around a MIMIC-derived profile schema; we re-wire the same persona taxonomy to drive the DDxPlus profile schema used in our other columns, leaving the persona axes themselves unchanged. Main experiments use the cooperative *Plain* persona; the four adversarial personas (*Overanxious*, *Distrustful*, *Dazed*, *Verbose*) are evaluated in the persona ablation of Appendix C.8.

Per-method doctor-side shims (MOBAYES engine patches, baseline reproductions, closed-set normalisation) are documented separately in Appendix C.14.

B.6 Implementation details

Table 6: Models used in experiments. All models accessed via API (OpenAI direct or OpenRouter). The patient simulator (GPT-5.4-nano) is shared across all experiments.

MOBAYES sensors	Standalone doctors
GPT-5.4-nano	GPT-5.4
Gemini 3.1 Flash Lite	Gemini 3.1 Pro
Llama-4-Scout	Llama-4-Maverick
GPT-OSS-20B	GPT-OSS-120B
Gemma 4 31B	Qwen 3.6 Plus
MiniMax M2.5	Kimi K2.5

All models are accessed via API. OpenAI models (GPT-5.4, GPT-5.4-nano) are called directly; all other models are routed through OpenRouter. MOBAYES components use temperature 0 for deterministic parsing and question generation.

C Extended experimental results

Unless stated otherwise, the ablations of Appendices use the DDxPlus 50-case set with the PatientSim wrapper.

C.1 LLM-elicited knowledge base, full results

This appendix expands on the LLM-generated knowledge base experiment reported in Section 5.3. In each ecosystem (GPT-5.4 and Gemini 3.1), the same frontier LLM populates the closed-disease KB and (separately) generates evaluation patients (Eq. 13). Both are noisy realisations of the LLM's beliefs; the remaining gap reflects how the LLM's own knowledge is deployed.

Headline finding. Across both ecosystems, an inexpensive MOBAYES sensor matches or approaches the frontier standalone doctor on top-1 accuracy at an order-of-magnitude lower per-token cost. A striking result is that the standalone doctor still drops sharply on *rare* diseases even though the patients are simulated from its own world model, while the Bayesian engine leverages the KB uniformly across prevalence. This is direct evidence that the gap is not informational: the LLM *has* the relevant probabilistic content (it generated the KB), but cannot deploy it consistently when forced to operate from natural language alone.

KB oracle row. For each ecosystem we report a *KB oracle* row that uses the generating LLM's own posterior over the KB as the diagnostic system. This is the upper bound implied by the KB itself; MOBAYES approaches it without the standalone doctor's inflation from training-data leakage. Both MOBAYES and the standalone are evaluated at $n{=}50$ on the Plain persona.

C.2 Prediction normalisation

All standalone doctor predictions undergo post-hoc normalisation to KB disease names via the MatchToKB procedure (Section B.3). The doctor never sees the KB disease list during the dialogue; normalisation is applied only after the doctor commits to a diagnosis. This ensures a fair comparison: the standalone doctor relies solely on its internal medical knowledge, while MOBAYES relies on the structured KB. The complementary direction, surfacing the KB disease list to the standalone doctor inside its system prompt, is reported separately as the Closed-World (CW) baseline in Table 21 (full protocol in Appendix C.14); mere KB-list awareness does not recover the architectural gap.

C.3 Confidence threshold experiments

Per-sensor τ^* is tuned by leave-one-out DHS-maximisation on the set used for that run; the chosen value is then applied to score the same set. Table 8 reports the DDXPlus confidence threshold sweep across all six MOBAYES sensor backends.

Table 7: LLM-generated KB experiments, full results. Each ecosystem uses the generating model's own clinical knowledge; patients sampled from the KB's likelihood model. n=50, Plain persona. Bottom block reports DHS stratified by LLM-estimated ED prevalence; the standalone doctor's drop on *rare* diseases (despite the patients being simulated from its own world model) is the architectural finding highlighted in Section 5.3.

	Accuracy		Selective Diagnosis		
System	**Top-1**	**Top-3**	**Sel. Acc**	**Cov.**	**DHS**
GPT-5.4 ecosystem					
KB oracle (c=1)	*76*	*88*	—	—	—
MoBayes + GPT-5.4-nano	70	84	78	90	83
SA GPT-5.4	72	80	81	86	84
SA GPT-5.4-nano	52	70	72	36	48
Gemini 3.1 ecosystem					
KB oracle (c=1)	*74*	*88*	—	—	—
MoBayes + Gemini FL	64	84	66	94	78
SA Gemini 3.1 Pro	70	78	73	96	83
SA Gemini 3.1 Flash Lite	62	76	63	92	75

	GPT-5.4 ecosystem			Gemini 3.1 ecosystem		
Prevalence	**MoBayes**	**SA Full**	**SA nano**	**MoBayes**	**SA Full**	**SA light**
Common	83	**95**	70	69	**88**	82
Rare	**84**	75	29	**82**	79	71

Table 8: Confidence threshold τ sweep on DDXPlus (n=50, Plain persona) for all MoBayes sensor backends. Sel. Acc. = selective Top-1 accuracy among committed cases; Cov. = fraction committed; DHS = harmonic mean of Sel. Acc. and Cov. (Eq. 1). Bold: DHS-optimal τ^* per sensor, used in Table 2.

	GPT-5.4-n			Gemini FL			Llama-4-Sc			GPT-OSS-20B			Gemma 4			MiniMax		
τ	SA	C	DHS	SA	C	DHS	SA	C	DHS	SA	C	DHS	SA	C	DHS	SA	C	DHS
0.00	78	100	88	74	100	85	56	100	72	61	100	76	64	100	78	60	100	75
0.10	78	100	88	74	100	85	56	100	72	61	100	76	64	100	78	60	100	75
0.20	**81**	**96**	**88**	**77**	**96**	**86**	57	98	72	61	100	76	64	100	78	60	100	75
0.30	81	94	87	77	94	84	58	96	73	**65**	**94**	**77**	**69**	**93**	**79**	**64**	**94**	**76**
0.40	83	92	87	78	92	85	**61**	**92**	**73**	71	84	77	71	86	78	69	78	73
0.50	82	90	86	79	86	82	62	84	71	70	76	73	77	79	78	71	76	73
0.60	81	86	84	81	74	77	64	78	70	74	71	73	81	75	78	77	68	72
0.70	80	80	80	88	66	75	73	66	69	79	67	73	80	71	76	79	66	72
0.80	86	72	78	88	64	74	74	62	68	87	61	72	84	68	75	83	58	68
0.90	85	68	76	93	60	73	85	54	66	90	59	71	83	64	73	89	52	66
0.95	88	64	74	97	58	73	85	52	65	90	59	71	83	64	73	88	50	64

LLM-generated KB threshold sweep. Table 9 reports the confidence threshold sweep for both LLM-generated KB ecosystems, including native and cross-KB transfer configurations. The optimal τ is selected by maximising DHS.

C.4 Question-selection policy ablations

Table 10 compares question-selection policies on DDXPlus (policy descriptions in Appendix A.2).

Focused top-k EIG improves accuracy over the global baseline by concentrating discriminative questions on the most probable diseases once the posterior begins to concentrate. The optimal configuration uses moderate focus strength applied early; too-strong focus degrades performance, likely because the model commits prematurely and stops exploring alternative hypotheses. Wider focus (k=5) is slightly worse than k=3, suggesting that a tighter candidate set produces more targeted questions. The main experiments (Table 2) use global EIG (λ=0) to keep the default policy simple and parameter-free; focused EIG is available as an optional enhancement.

Table 9: Confidence threshold sweep for LLM-generated KB experiments (n=50, Plain persona). Bold: DHS-optimal operating point. Cross-KB rows evaluate on patients generated from the other ecosystem's KB.

	GPT KB (native)			**GPT KB → Gem. pat.**			**Gem. KB (native)**			**Gem. KB → GPT pat.**		
τ	SA	Cov	DHS	SA	Cov	DHS	SA	Cov	DHS	SA	Cov	DHS
0.00	70	100	82	52	100	68	64	100	78	42	100	59
0.10	70	100	82	52	100	68	64	100	78	42	100	59
0.20	73	96	83	**54**	**96**	**69**	64	100	**78**	42	100	59
0.30	**78**	**90**	**83**	61	76	67	66	94	78	**48**	**84**	**61**
0.40	83	72	77	70	60	65	79	76	77	57	56	57
0.50	89	52	66	68	56	61	84	62	71	67	42	52
0.60	96	44	60	68	50	58	88	50	64	73	30	43
0.70	100	40	57	68	44	54	90	40	55	75	24	36
0.80	100	38	55	77	34	47	88	32	47	80	20	32
0.90	100	32	49	80	30	44	100	16	28	78	18	29

Table 10: Question-selection policy ablation on DDXPlus (MOBAYES + GPT-5.4-nano, Plain persona). *Global EIG* selects the most informative question across all diseases; *Focused top-k EIG* adds a discrimination term over the k most probable diseases, weighted by λ, activated once $\max_d b_t(d) \geq \theta$. DHS uses DHS-optimal τ^*. The main experiments (Table 2) use Global EIG; focused EIG can improve DHS by up to +7 pp. Colour intensity: green = higher, red = lower.

		Accuracy		**Selective Diagnosis**		
Policy	**Params**	**Top-1**	**Top-3**	**Sel. Acc**	**Cov.**	**DHS**
Global EIG (baseline)	λ=0	76	88	83	92	87
Focused top-3 EIG	λ=0.3, θ=0.3	80	88	87	92	89
Focused top-3 EIG	λ=0.5, θ=0.3	**88**	**92**	**96**	92	**94**
Focused top-3 EIG	λ=0.3, θ=0.5	84	88	88	96	92
Focused top-3 EIG	λ=1.0, θ=0.3	72	80	86	84	85
Focused top-5 EIG	λ=0.3, θ=0.3	84	**92**	88	96	92

C.5 Prior distribution sensitivity

We compare three prior initialisation strategies $b_0(d)$ on DDXPlus (n=50, MOBAYES + GPT-5.4-nano):

- **Empirical.** $b_0(d) = \hat{p}(d)$, the maximum-likelihood prevalence estimated from the DDXPlus training split.
- **Uniform.** $b_0(d) = 1/K$, assigning equal mass to all K diseases.
- **Conditional.** $b_0(d \mid a, s) \propto \hat{p}(d, a, s)$, prevalence stratified by age group a and sex s (six demographic bins), estimated from training counts.

Table 11 reports results at each prior's DHS-optimal threshold.

Table 11: Prior distribution ablation on DDXPlus (n=50, MOBAYES + GPT-5.4-nano). DHS-optimal τ^* selected per prior. Colour intensity: green = higher, red = lower.

	Accuracy		**Selective Diagnosis**		
Prior	**Top-1**	**Top-3**	**Sel. Acc**	**Cov.**	**DHS**
Empirical	70	88	71	98	83
Uniform	70	92	71	98	83
Conditional	**74**	90	**79**	94	**86**

Empirical and uniform priors achieve identical DHS despite assigning very different initial mass to diseases. This insensitivity is expected given the minimum-question threshold $T_{\min}$=12: the engine collects enough evidence to wash out the prior before making a decision. The conditional prior achieves the highest DHS by starting closer to the true posterior for each demographic group, reducing the evidence burden.

As the disease space scales and symptom overlap increases, however, prior choice is likely to matter more: with many competing hypotheses sharing similar features, a well-calibrated prior can break ties that evidence alone cannot resolve within the question budget.

Per-prevalence breakdown. Table 12 stratifies the empirical vs. uniform comparison by prevalence tercile (Appendix C.9).

Table 12: Effect of prior choice on Top-1 accuracy by disease prevalence (n=50, MoBayes + GPT-5.4-nano). Diseases ranked by training-set prevalence and split into terciles. Δ: uniform − empirical.

Prevalence group	n	Empirical	Uniform	Δ
Common (top 16)	20	70	70	±0
Medium (mid 17)	17	**76**	65	−11
Rare (bottom 16)	13	62	**77**	+15
All	50	70	70	±0

The uniform prior improves accuracy on rare diseases at the cost of medium-prevalence diseases. This trade-off arises because the empirical prior assigns disproportionately high mass to common diseases, which helps conditions sharing features with frequent diagnoses but hinders rare diseases that must overcome a strong prior disadvantage. Under the uniform prior, the posterior is driven entirely by evidence, benefiting rare diseases whose distinctive symptom profiles are otherwise masked by prior dominance.

C.6 Accuracy-coverage analysis

A selective classifier's quality is best assessed not at a single threshold but across the full accuracy–coverage frontier. Figure 9 plots selective accuracy as a function of coverage for all six MoBayes sensors on DDXPlus, with standalone doctors shown as fixed points.

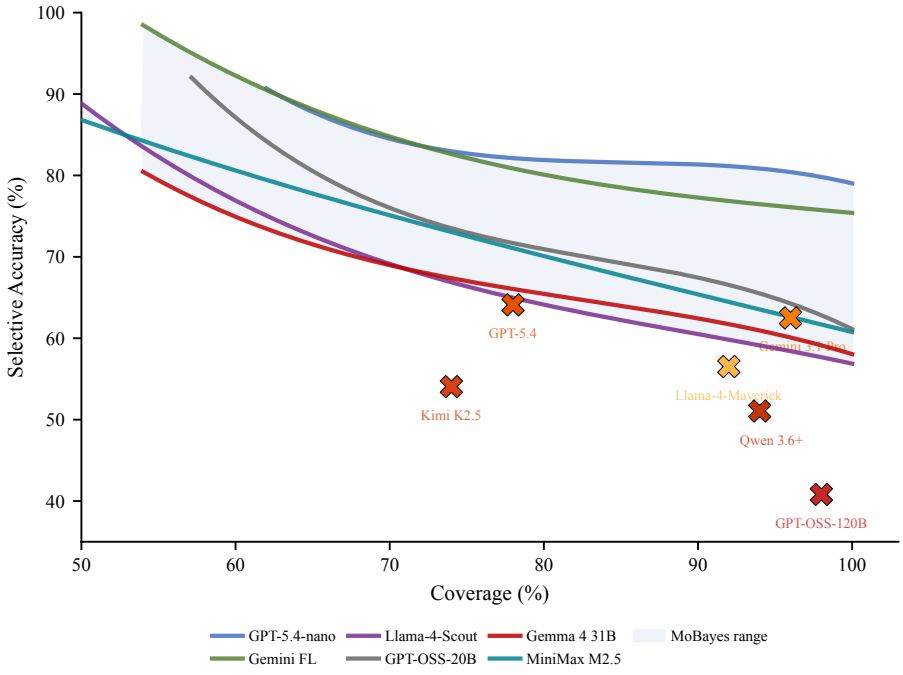


Figure 9: DDXPlus selective accuracy vs. coverage (n=50). Each session runs to completion; the curve is traced by sweeping the commit threshold τ on the final posterior (smoothed via spline interpolation). Standalone doctors (×) operate at a single, abstention-determined point. MoBayes dominates at every matched coverage level.

At matched coverage, every MOBAYES sensor dominates every standalone doctor: the MOBAYES envelope lies entirely above the cluster of standalone points. This gap widens as coverage decreases, because MOBAYES's explicit posterior allows it to selectively abstain on genuinely uncertain cases, whereas standalone doctors lack a reliable confidence signal for threshold-based abstention. The result is that MOBAYES offers strictly better accuracy at every operating point a deployer might choose.

C.7 Computational cost analysis

Table 13 reports per-model API pricing and estimated per-patient cost on DDXPlus. The per-token view of the cost–accuracy tradeoff is reported in Figure 3 (main text); the per-patient view, which folds in dialogue length and per-turn token routing, is shown in Figure 10 below.

All API calls in our experiments are routed through CacheSaver (47), a namespace-aware list-valued inference cache that deduplicates repeated queries while preserving i.i.d. statistical integrity within a namespace, reducing the marginal compute and carbon cost of repeated evaluations and stabilising reproducibility across runs.

Table 13: API pricing and per-patient cost on DDXPlus. Listed prices as of April 2026 from each model's OpenRouter page (OpenAI models via direct API). Actual cost may vary by provider and routing. Cost/pat is the mean estimated spend per diagnostic session. Colour intensity: green = better, red = worse.

	Pricing ($/M)		**Performance**		
System	**In**	**Out**	**Sel. Acc**	**DHS**	**Cost/pat**
MOBAYES + *sensor LLM (ours)*					
+ GPT-5.4-nano	0.20	1.25	**81**	**88**	1.4
+ Gemini 3.1 FL	0.25	1.50	77	86	1.6
+ GPT-OSS-20B	0.03	0.14	70	78	0.9
+ MiniMax M2.5	0.12	0.99	64	76	2.0
+ Gemma 4 31B	0.13	0.38	60	74	0.9
+ Llama-4-Scout	0.08	0.30	61	73	1.0
Standalone LLM doctors					
SA GPT-5.4	2.50	15.00	64	70	19.6
SA Gemini 3.1 Pro	2.00	12.00	63	76	25.0
SA Qwen 3.6+	0.33	1.95	51	66	4.2
SA Kimi K2.5	0.38	1.72	54	63	2.7
SA GPT-OSS-120B	0.04	0.19	41	58	1.0
SA Llama-4-Maverick	0.15	0.60	57	70	0.9

Cost/pat in US cents per diagnostic session.

C.8 Persona robustness

A clinically useful system must handle diverse patient communication styles. We evaluate robustness across five patient personas that vary personality, language proficiency, recall accuracy, and cognitive state (Table 14).

We test MOBAYES and standalone GPT-5.4-nano across four adversarial personas in addition to the cooperative baseline. Table 15 reports the complete breakdown.

MOBAYES maintains high coverage and stable DHS across all adversarial personas, degrading only moderately from the plain baseline. In contrast, the standalone doctor's coverage collapses under adversarial conditions: when patients are evasive or confused, the model abstains on the majority of cases, producing low DHS despite sometimes achieving high selective accuracy on the few cases it does commit to. The MOBAYES advantage widens under adversarial conditions because the Bayesian backbone accumulates evidence independently of the patient's communication style, even noisy or incomplete responses contribute partial information through Pearl-style soft evidence.

The distrustful persona yields the highest MOBAYES DHS because both systems achieve comparable selective accuracy on committed cases, the difference is coverage: the standalone doctor becomes

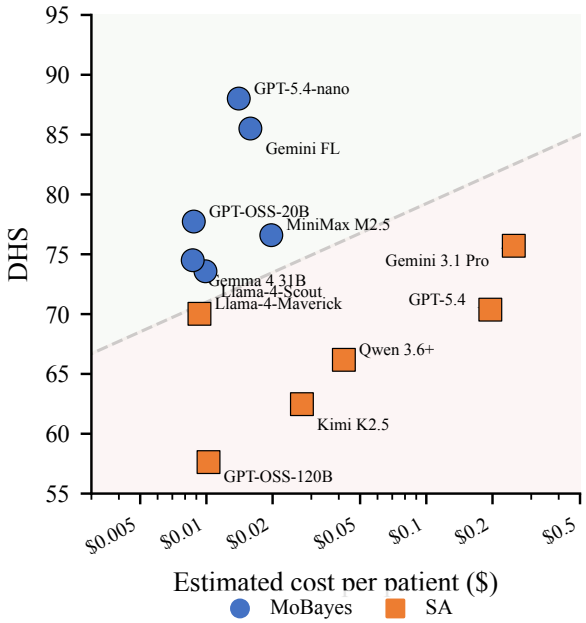


Figure 10: DHS vs. estimated per-patient API cost on DDXPlus (n=50). MoBayes sensors (circles) achieve higher DHS than frontier standalone doctors (squares) at 10–25$\times$ lower per-patient cost. Cost is the mean over 50 sessions of doctor-side and patient-simulator token spend, summing input and output tokens at each model's listed rate (Table 13); patient simulator pricing is fixed at GPT-5.4-nano ($0.20/M in, $1.25/M out).

Table 14: Patient personas used for robustness evaluation. Each persona varies the simulated patient along four axes from PatientSim (43).

Persona	Proficiency	Recall	Confusion	Personality	Effect on dialogue
Plain	B	High	None	Cooperative	Baseline. Answers directly.
Overanxious	B	High	None	Anxious	Over-reports symptoms, adds false positives.
Distrustful	B	High	None	Skeptical	Withholds information, terse answers.
Dazed	A	Low	Moderate	Confused	Inconsistent responses across turns.
Verbose	B	Low	None	Talkative	Buries relevant info in tangential detail.

too uncertain to commit on most patients, while MoBayes still accumulates sufficient posterior confidence through Bayesian updating even from partial evidence.

Figure 11 visualises the consistent gap across all personas.

C.9 Accuracy by disease prevalence

A clinically important question is whether the system maintains accuracy across the full prevalence spectrum. We stratify the 49 DDXPlus diseases into three groups by rank-ordering their population prevalence and splitting into equal-sized terciles: *common* (top 16), *medium* (middle 17), and *rare* (bottom 16).

Table 16 reports Top-1 accuracy, selective accuracy, coverage, and DHS for the strongest MoBayes configuration (GPT-5.4-nano) and the strongest standalone doctor (SA GPT-5.4).

MoBayes maintains a nearly flat DHS profile across the prevalence spectrum, whereas the standalone doctor degrades sharply on rare diseases. This gap arises from two complementary effects. First, the

Table 15: Per-persona results on DDXPlus (GPT-5.4-nano as sensor/doctor). MoBayes uses DHS-optimal τ^*; SA uses self-determined abstention. Δ: MoBayes $-$ SA. Colour intensity: green = higher, red = lower.

	MoBayes			Standalone			
Persona	**Sel. Acc**	**Cov.**	**DHS**	**Sel. Acc**	**Cov.**	**DHS**	**ΔDHS**
Plain	81	96	88	64	78	70	+18
Overanxious	79	96	87	80	40	53	+34
Distrustful	88	96	92	89	36	51	+41
Dazed	71	96	82	50	48	49	+33
Verbose	72	100	84	56	64	60	+24
Mean (adv.)	*78*	*97*	*86*	*69*	*47*	*53*	*+33*

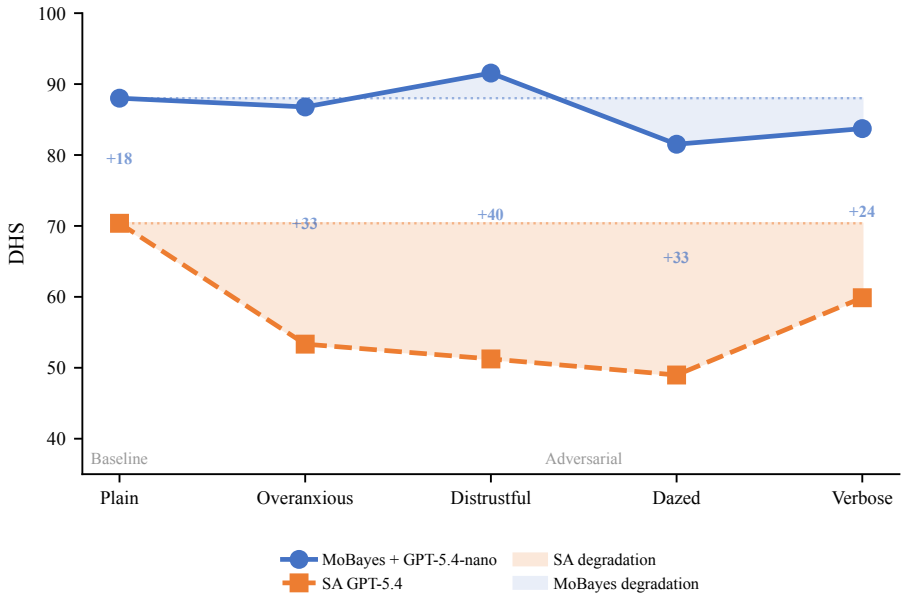


Figure 11: DHS across patient personas. Shaded areas show degradation from the plain baseline (dotted lines). MoBayes (blue) remains near its baseline under all adversarial conditions; the standalone doctor (orange) collapses. The contrast between the two shaded areas illustrates the robustness advantage of structured Bayesian reasoning over end-to-end LLM inference.

Bayesian engine conditions on a knowledge base whose likelihood tables are prevalence-agnostic: the Dirichlet counts encode symptom–disease associations regardless of how common a disease is. Second, the standalone doctor inherits the frequency bias of its LLM training corpus, diseases that appear more often in medical text receive disproportionate probability mass, systematically disadvantaging rare conditions.

The coverage column reinforces this distinction. MoBayes rarely abstains regardless of prevalence, while the standalone doctor abstains more aggressively on rare cases yet still achieves lower accuracy among the cases it does diagnose. In clinical terms, the Bayesian architecture provides both higher reliability and higher willingness to commit on the long tail of the disease distribution.

C.10 Failure mode analysis

On DDxPlus 50 cases (PatientSim), we classify each committed misdiagnosis (confidence $\geq \tau^*$, top-1 incorrect) along three independent, non-exclusive axes.

1. **KB Failure.** We run an *oracle* test: all ground-truth features are supplied at confidence c=1.0. We then measure the posterior gap between the oracle's top-1 and top-2 diseases. If this gap falls below a threshold γ (γ=0.80), the KB cannot reliably discriminate the disease pair.

Table 16: Diagnostic performance stratified by disease prevalence tercile on DDXPlus (n=50). MOBAYES uses DHS-optimal τ^*; SA uses self-determined abstention. Diseases rank-ordered by population prevalence and split into equal-sized groups.

				Selective Diagnosis		
Prevalence	**Model**	**Top-1**	**Top-3**	**Sel. Acc**	**Cov.**	**DHS**
Common (n=20)	MOBAYES + nano	**90**	**100**	**90**	**100**	**95**
	SA GPT-5.4	55	85	60	75	67
Medium (n=17)	MOBAYES + nano	71	88	75	**94**	**84**
	SA GPT-5.4	71	88	**79**	82	80
Rare (n=13)	MOBAYES + nano	**69**	77	**75**	**92**	**83**
	SA GPT-5.4	54	77	50	77	61
All (n=50)	MOBAYES + nano	**78**	**90**	**81**	**96**	**88**
	SA GPT-5.4	60	84	64	78	70

2. **LLM Failure.** The LLM pipeline (verbaliser + patient simulator + parser) injected incorrect evidence into the engine. Two subtypes:
   - *False Positive (FP)*: the engine asks about a feature *absent* from the patient's ground-truth profile; the pipeline returns `yes`. If more than 2 such turns occur in a session, the case is flagged. The threshold reflects the empirical separation between successful sessions (mean 1.3 FP) and failure sessions (mean 3.5 FP).
   - *Wrong Evidence (WE)*: the engine asks about a feature *present* in the ground-truth profile, but the extracted value contradicts the ground truth. Any occurrence triggers the flag.

   This taxonomy is *conservative*: it captures only detectable errors. Cases where the parser maps an uncertain response to `no` for an absent feature are not flagged, nor are confidence miscalibrations.
3. **Inference Failure.** The KB is adequate and the evidence pipeline introduced no detectable errors, yet the engine converged to the wrong diagnosis. Two subtypes:
   - *Close*: the ground truth remains in the top-3 posterior at session end, but the question budget or EIG policy did not resolve the differential.
   - *Diverged*: the ground truth is not in the top-3. The engine committed confidently to a wrong hypothesis.

Results. Table 17 reports the analysis for the six MOBAYES sensors (committed failures only, τ^* per sensor).

Table 17: Failure mode analysis on DDXPlus (committed misdiagnoses only). Axes are non-exclusive: a case may trigger multiple failure modes. Colour intensity: red = more failures.

			LLM		Inference	
Sensor	**Fail**	**KB**	**FP**	**WE**	**Close**	**Div.**
GPT-5.4-nano	9	1	6	0	2	0
Gemini FL	11	2	4	1	2	2
Llama-4-Scout	17	2	6	2	3	4
GPT-OSS-20B	13	4	5	0	2	2
Gemma 4 31B	18	4	11	0	1	2
MiniMax M2.5	15	3	9	1	2	0
Total	**83**	**16**	**41**	**4**	**12**	**10**
%		19%	49%	5%	14%	12%

KB failures are patient-specific, not sensor-specific. The KB failure count varies across sensors (1–4) not because the KB changes, but because stronger sensors resolve cases that weaker ones cannot. The same patients trigger oracle failure regardless of sensor. However, a capable sensor may still overcome a marginal KB gap through better evidence collection, while a weaker sensor fails on those cases and gets counted.

LLM failure dominates and scales with sensor quality. The LLM pipeline accounts for the majority of committed errors, with false positives as the dominant subtype. Critically, this failure mode scales with sensor capability: frontier sensors produce substantially fewer false positives than smaller models (Table 17). This suggests that scaling the sensor LLM, without changing the Bayesian engine, directly reduces the dominant failure mode.

Inference failures confirm the architectural thesis. The strongest sensors have zero diverged inference failures: every committed error is either a KB limitation or an evidence collection problem. When the LLM pipeline delivers accurate evidence, the Bayesian engine reliably converges to the correct diagnosis. Weaker sensors produce noisier evidence that steers the EIG policy toward unproductive questions, causing the engine to commit to wrong hypotheses even when the KB contains sufficient information.

C.11 Disease scaling analysis

We evaluate how diagnostic performance scales with the size of the candidate disease space. Random subsets of $K \in \{10, 20, 30, 40\}$ diseases are sampled from DDXPlus, and one patient per disease is evaluated for both MoBayes and the standalone doctor.

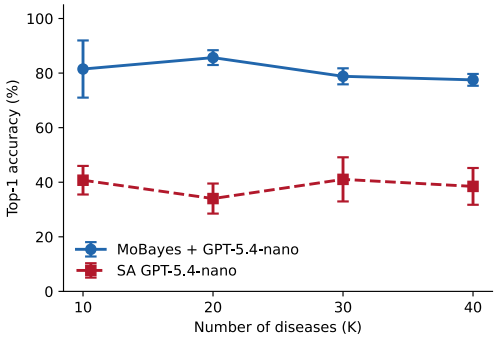


Figure 12: Top-1 accuracy vs. KB size K. MoBayes remains stable across a $4\times$ increase in disease space; the standalone doctor is flat regardless of K.

Table 18: Disease scaling: Top-1 accuracy on DDXPlus random subsets (3 seeds, $N = K$ patients). Δ: MoBayes $-$ SA.

K	MoBayes	SA	Δ
10	82	41	+41
20	86	34	+52
30	79	41	+38
40	78	39	+39

MoBayes maintains stable accuracy as the disease space quadruples, because its inference complexity scales with the KB structure rather than with the LLM's reasoning capacity. The standalone doctor, by contrast, operates over its own implicit disease space regardless of K—it does not know how many candidates the KB contains—and its performance remains flat. Providing the candidate list explicitly would likely degrade the standalone doctor further at higher K, as a longer list introduces confusion without a structured mechanism to leverage it.

C.12 Diagnostic session walkthroughs

Figure 13 illustrates the engine's belief dynamics on a representative case: competing hypotheses rise and fall as evidence accumulates, and the posterior of the ground-truth disease converges to near-certainty.

We present an example session below.

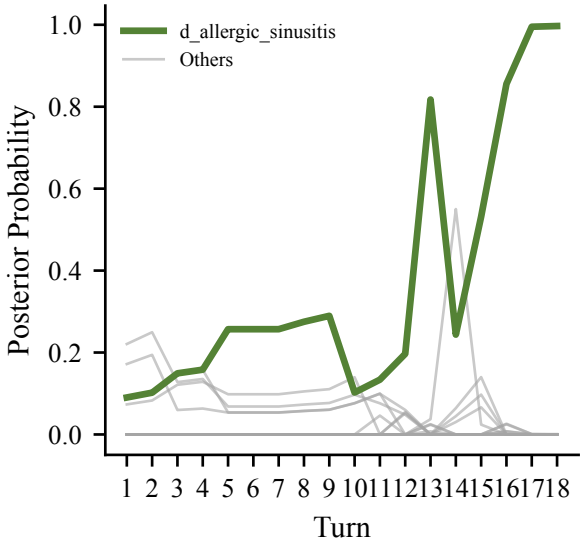


Figure 13: Posterior trajectory for a single session (Allergic sinusitis, GPT-5.4-nano). The ground-truth disease (green) starts at 9%, competes with alternatives, and converges to near-certainty.

Case 1: Spontaneous pneumothorax

Correct, confidence = 0.999

"I have chest pain even at rest, upper chest pain, and pleuritic chest pain"

T1 Dr: Does your thoracic pain spread or radiate to another area, such as your chest, ribs, shoulder, or around to the front of your body?
Pt: *"No. My pain is in my upper chest and both breasts, and it doesn't spread or radiate to other areas."*

T2 Dr: Do you have chronic obstructive pulmonary disease (COPD), such as chronic bronchitis or emphysema?
Pt: *"No. I don't have COPD or chronic bronchitis/emphysema that I know of."*

T3 Dr: What color is the rash? For example, is it red, pink, dark, pale, or yellow, or is there no rash?
Pt: *"There is no rash."*

T4 Dr: Do you feel any pain in the upper middle part of your abdomen (the area just below your breastbone)?
Pt: *"No. My pain is in my upper chest area, not in the upper middle abdomen below the breastbone."*

T5 Dr: Have you had any surgery or a procedure that required anesthesia within the last month?
Pt: *"No. I haven't had any surgery or anesthesia within the last month."*

T6 Dr: Do you feel your heart beating fast, irregularly, or having palpitations right now?
Pt: *"No. I don't feel my heart racing or having palpitations right now."*

T7 Dr: Do you currently smoke cigarettes?
Pt: *"Yes. I currently smoke cigarettes."*

T8 Dr: Do you consider yourself significantly overweight compared to other people of the same height?
Pt: *"No. I don't consider myself significantly overweight."*

T9 Dr: Do you constantly feel fatigued, or do you have sleep that doesn't leave you feeling rested?
Pt: *"No. I'm not constantly fatigued, and I don't have sleep that never leaves me rested."*

T10 Dr: Do you drink alcohol excessively, or do you feel you have an addiction to alcohol?
Pt: *"No. I don't drink excessively and I don't feel addicted to alcohol."*

T11 Dr: Have you ever smoked cigarettes in the past, but you don't smoke now (a former smoker)?
Pt: *"No. I'm still smoking cigarettes now, so I'm not a former smoker."*

T12 Dr: When you feel the pain, does it feel like a tugging or pulling sensation?
Pt: *"No. It feels more like a knife-stroke, heartbreaking pain, not tugging or pulling."*

T13 Dr: Does your pain feel exhausting, like it wears you out or drains your energy?
Pt: *"No. It's very severe, but it doesn't feel exhausting or like it drains my energy."*

T14 Dr: Have you ever had a spontaneous pneumothorax (a collapsed lung that happened without a clear injury or accident)?
Pt: *"Yes. I've had a spontaneous pneumothorax before."*

T15 Dr: Is your pain heartbreaking or emotionally overwhelming in a way that feels hard to bear?
Pt: *"Yes. It's heartbreaking and hard to bear."*

Posterior (Sp. pneumo)

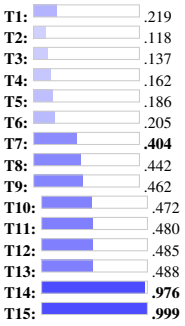


Table 19: Bootstrap 95% CIs for Table 2 (frontier standalone vs. MOBAYES, n=50 per cell). $B = 2000$; cells show point [2.5%, 97.5%]. DHS coloured per benchmark column.

	DDxPlus PatientSim			DDxPlus HT-sim			AgentClinic-MedQA		
Method	T-1	SA	DHS ↑	T-1	SA	DHS ↑	T-1	SA	DHS ↑
Standalone LLM doctors (frontier)									
SA GPT-5.4	60.0 [46.0,72.0]	64.1 [50.0,78.9]	70.4 [59.3,79.8]	58.0 [44.0,72.0]	76.0 [57.1,92.0]	60.3 [48.0,70.9]	60.0 [46.0,74.0]	62.8 [47.7,77.3]	72.6 [61.5,82.1]
SA Gemini 3.1 Pro	60.0 [46.0,74.0]	62.5 [49.0,75.5]	75.7 [64.8,85.3]	60.0 [46.0,74.0]	73.2 [59.0,85.7]	77.3 [66.9,85.9]	66.0 [54.0,80.0]	68.8 [55.1,81.6]	80.1 [70.1,88.9]
SA Llama-4-Maverick	56.0 [42.0,70.0]	56.5 [41.7,70.5]	70.0 [57.5,80.4]	46.0 [32.0,60.0]	56.0 [36.4,75.0]	52.8 [39.8,63.2]	56.0 [42.0,70.0]	58.3 [44.7,72.9]	72.6 [60.6,82.9]
SA Qwen 3.6 Plus	48.0 [34.0,62.0]	51.1 [36.7,64.6]	66.2 [52.2,76.9]	60.0 [46.0,74.0]	65.0 [50.0,79.0]	71.7 [60.3,80.9]	64.0 [50.0,76.0]	63.3 [50.0,76.0]	76.9 [65.8,85.5]
SA Kimi K2.5	42.0 [28.0,56.0]	54.1 [37.5,71.0]	62.5 [49.2,73.2]	42.0 [28.0,56.0]	40.7 [23.1,60.7]	46.4 [31.6,58.3]	54.0 [40.0,68.0]	57.4 [43.5,71.4]	71.3 [59.1,81.5]
SA GPT-OSS-120B	40.0 [26.0,54.0]	40.8 [27.1,55.1]	57.6 [42.2,70.5]	54.0 [40.0,68.0]	54.5 [37.5,71.0]	59.7 [46.9,70.2]	48.0 [34.0,62.0]	49.0 [34.7,63.0]	65.3 [51.2,76.5]
MOBAYES + *sensor LLM*									
+ GPT-5.4-nano	78.0 [66.0,90.0]	81.2 [68.9,91.7]	88.0 [79.8,94.7]	80.0 [68.0,90.0]	83.3 [72.0,93.6]	89.2 [82.0,94.8]	76.0 [64.0,88.0]	79.2 [67.3,89.8]	86.8 [79.0,93.6]
+ Gemini 3.1 Flash Lite	74.0 [62.0,86.0]	77.1 [64.6,88.0]	85.5 [77.2,92.6]	84.0 [74.0,94.0]	85.7 [75.5,94.0]	91.4 [85.3,96.9]	78.0 [66.0,88.0]	86.7 [76.6,95.7]	88.3 [81.1,93.8]
+ GPT-OSS-20B	60.0 [46.0,74.0]	68.2 [54.5,81.4]	76.8 [66.6,85.7]	50.0 [36.0,64.0]	61.0 [46.3,75.6]	69.9 [58.0,79.5]	50.0 [36.0,64.0]	58.1 [43.2,73.2]	69.4 [57.2,79.2]
+ MiniMax M2.5	60.0 [46.0,74.0]	66.7 [53.3,80.0]	76.6 [66.3,85.3]	60.0 [46.0,74.0]	75.0 [61.5,88.1]	77.4 [67.8,85.9]	50.0 [36.0,64.0]	52.1 [37.5,67.3]	67.5 [53.9,79.0]
+ Gemma 4 31B	58.0 [44.0,72.0]	61.7 [47.8,75.5]	74.5 [62.9,84.0]	82.0 [70.0,92.0]	83.7 [72.9,93.8]	90.3 [82.9,95.8]	64.0 [50.0,78.0]	65.3 [52.0,79.2]	78.4 [67.5,87.6]
+ Llama-4-Scout	56.0 [42.0,70.0]	60.9 [46.7,75.0]	73.3 [61.5,83.1]	80.0 [68.0,90.0]	87.0 [76.6,95.7]	89.4 [82.1,94.9]	68.0 [56.0,80.0]	79.1 [66.7,90.9]	82.4 [73.6,89.4]

Table 20: Bootstrap 95% CIs and token totals for Table 21 (combined n=100). $B = 2000$; cells show point [2.5%, 97.5%]. Tok/case is the average doctor + patient prompt+completion tokens per diagnostic conversation.

	DDxPlus PatientSim			DDxPlus HT-sim			AgentClinic-MedQA		
Method	T-1	DHS ↑	Tok/case	T-1	DHS ↑	Tok/case	T-1	DHS ↑	Tok/case
Prompt-based									
Bayesian-CoT	60.0 [51.0,69.0]	74.0 [66.1,80.8]	46.6K	41.0 [31.0,51.0]	44.5 [34.7,52.8]	34.1K	63.0 [53.0,73.0]	79.3 [72.7,85.4]	37.8K
Chain-of-Thought (CoT)	52.0 [42.0,62.0]	69.0 [60.6,76.6]	30.8K	41.0 [31.0,51.0]	29.2 [19.1,38.4]	15.8K	71.0 [62.0,80.0]	84.1 [78.4,89.3]	22.5K
Closed-World (CW)	40.0 [31.0,49.0]	55.4 [45.8,63.2]	32.0K	45.0 [35.0,55.0]	20.7 [11.0,30.2]	23.9K	40.0 [31.0,50.0]	62.3 [54.2,69.8]	25.7K
DDx-CoT	42.0 [32.0,52.0]	54.6 [44.9,62.4]	28.8K	37.0 [28.0,46.0]	16.1 [7.4,25.3]	18.5K	40.0 [31.0,50.0]	61.8 [53.5,69.6]	23.7K
Information-pursuit									
UoT	28.0 [19.0,37.0]	43.8 [31.9,54.0]	37.7K	5.0 [1.0,9.0]	9.5 [2.0,16.5]	28.2K	38.0 [29.0,47.0]	56.1 [45.7,65.3]	34.6K
Agent-based									
MEDDxAgent (paper-best)	45.0 [35.0,55.0]	62.1 [51.9,71.0]	108.3K	56.0 [46.0,66.0]	71.8 [63.0,79.5]	88.1K	71.0 [62.0,80.0]	83.0 [76.5,88.9]	106.8K
MEDDxAgent (closed-world abl.)	41.0 [31.0,51.0]	58.2 [47.3,67.5]	79.9K	39.0 [30.0,49.0]	56.1 [46.2,65.8]	71.9K	80.0 [72.0,88.0]	88.9 [83.7,93.6]	69.0K
MediQ-Expert (force-final)	43.0 [33.0,52.0]	60.1 [49.6,68.4]	39.3K	34.0 [25.0,43.0]	50.7 [40.0,60.1]	41.9K	76.0 [67.0,84.0]	86.4 [80.2,91.3]	30.0K
AgentClinic	27.0 [19.0,36.0]	52.0 [42.8,59.9]	18.1K	28.0 [19.0,37.0]	52.0 [41.9,60.2]	29.7K	66.0 [56.0,75.0]	79.7 [72.2,85.8]	6.2K
MediQ-Expert (strict abstain)	32.0 [23.0,41.0]	56.2 [46.7,64.0]	39.3K	25.0 [17.0,34.0]	49.6 [40.2,57.4]	41.9K	58.0 [48.0,68.0]	75.7 [68.4,82.1]	30.0K
Fine-tuned (specialised model)									
CoD / DiagnosisGPT-34B	11.0 [5.0,18.0]	19.8 [9.5,30.5]	13.4K	4.0 [1.0,8.0]	10.7 [2.6,20.3]	12.7K	46.0 [36.0,56.0]	67.2 [58.9,74.1]	31.2K
Ours									
MOBAYES (gpt-5.4-nano)	82.0 [74.0,89.0]	90.3 [85.6,94.3]	16.4K	81.0 [73.0,88.0]	90.0 [85.3,93.8]	18.0K	70.0 [61.0,79.0]	83.1 [77.1,88.5]	18.9K

C.13 Bootstrap confidence intervals

To quantify the uncertainty in the point estimates, we compute bootstrap CIs (B=2000, percentile method) for every cell of Tables 2 and 21. Cell entries are point $_{[2.5\%,\,97.5\%]}$. For MOBAYES, τ^* is locked at the LOOCV value once on the original data (per benchmark, per half for the combined n=100 table) and held fixed across replicates. Tables 19 and 20 report the full CIs.

Combined point estimates with category-internal DHS ranking. Table 21 reports the same combined-n=100 comparison as Table 20 but as point estimates, broken out into all five metrics (T-1, T-3, SA, Cov, DHS), with rows ranked by average DHS within each category.

C.14 Baseline protocols and adaptations

Case selection. AgentClinic-MedQA cases are drawn from the 213-case pool at a fixed seed, weighted by disease frequency, with ties resolved at random. As in MEDDxAgent (14), headline comparisons and significance tests use 100 cases per benchmark.

Per-method configuration. All non-fine-tuned methods use gpt-5.4-nano. Released code is run under each paper's recommended best configuration; only the scientifically substantive parameters are listed.

Table 21: Combined n=100 baseline comparison: canonical n=50 plus a disjoint n=50 replication. Per Appendix A.4, MOBAYES's threshold τ^* is LOOCV-tuned on each 50-case set; combined SA, Cov, DHS aggregate committed/correct counts across halves rather than re-tuning on n=100. T-k is naive (independent of commitment); SA, Cov, DHS are commit-based. DHS coloured per benchmark column; rows ranked within each category by average DHS.

	DDxPlus 100 (MOBAYES PatientSim)					**DDxPlus 100 (MEDDxAgent HT-sim)**					**AgentClinic-MedQA 100**				
Method	T-1	T-3	SA	Cov.	DHS ↑	T-1	T-3	SA	Cov.	DHS ↑	T-1	T-3	SA	Cov.	DHS ↑
Prompt-based															
DDx-CoT	42.0	75.0	50.8	59.0	54.6	37.0	54.0	77.8	9.0	16.1	40.0	75.0	78.4	51.0	61.8
Closed-World (CW)	40.0	64.0	50.0	62.0	55.4	45.0	59.0	75.0	12.0	20.7	40.0	73.0	75.5	53.0	62.3
Chain-of-Thought (CoT)	52.0	71.0	54.8	93.0	69.0	41.0	59.0	63.2	19.0	29.2	71.0	83.0	**88.8**	80.0	84.1
Bayesian-CoT	60.0	75.0	62.0	92.0	74.0	41.0	55.0	58.3	36.0	44.5	63.0	77.0	76.8	82.0	79.3
Information-pursuit															
UoT	28.0	41.0	28.0	**100.0**	43.8	5.0	9.0	5.0	**100.0**	9.5	38.0	52.0	39.6	96.0	56.1
Agent-based															
MediQ-Expert (strict abstain)	32.0	–	50.8	63.0	56.2	25.0	–	56.8	44.0	49.6	58.0	–	85.3	68.0	75.7
AgentClinic	27.0	29.0	50.9	53.0	52.0	28.0	29.0	43.8	64.0	52.0	66.0	68.0	66.7	99.0	79.7
MediQ-Expert (force-final)	43.0	–	43.0	**100.0**	60.1	34.0	–	34.0	**100.0**	50.7	76.0	–	76.0	**100.0**	86.4
MEDDxAgent (closed-world abl.)	41.0	68.0	41.0	**100.0**	58.2	39.0	60.0	39.0	**100.0**	56.1	**80.0**	85.0	80.0	**100.0**	**88.9**
MEDDxAgent (paper-best)	45.0	73.0	45.0	**100.0**	62.1	56.0	75.0	56.0	**100.0**	71.8	71.0	**88.0**	71.0	**100.0**	83.0
Fine-tuned (specialised model)															
CoD / DiagnosisGPT-34B	11.0	17.0	11.0	**100.0**	19.8	4.0	7.0	5.8	69.0	10.7	46.0	46.0	78.0	59.0	67.2
Ours															
MOBAYES (gpt-5.4-nano)	**82.0**	**92.0**	**84.5**	97.0	**90.3**	**81.0**	**92.0**	**87.1**	93.0	**90.0**	70.0	84.0	74.5	94.0	83.1

- **Closed-World prompt enrichment.** CW is not a separate method but a prompt-level addendum: the standalone doctor's system prompt is augmented with the active closed disease list and an instruction to prefer names from this list at commit time. Including CW measures whether merely surfacing the candidate space is sufficient on top of an unconstrained standalone doctor.
- **Chain-of-Thought (CoT) (29).** Prompt augmented with the standard zero-shot Chain-of-Thought instruction: each turn the model emits a step-by-step rationale that summarises findings, lists top candidates with supporting and refuting evidence, and identifies the next discriminating action.
- **DDx-CoT (45).** Prompt augmented with the Differential Diagnosis Chain-of-Thought instruction of Savage et al.: each turn the model builds or refines a differential and lists supporting and refuting findings per candidate before choosing the next question. The instruction is taken verbatim; the source's single-turn worked examples are omitted because we evaluate in a multi-turn dialogue setting.
- **Bayesian-CoT (45).** Prompt augmented with Savage et al.'s Bayesian Inference CoT instruction: each turn the model is asked to state an explicit posterior over top candidates and update it with new findings using prior plus likelihood-ratio reasoning. This mirrors MOBAYES's Bayesian engine in instruction form, so the comparison isolates whether prompt-level Bayesian reasoning is sufficient or whether the deterministic engine is needed.
- **MediQ-Expert (13).** The paper's recommended ScaleExpert variant with rationale generation, self-consistency n=3, an abstention threshold of at least 4 on a 1 to 5 Likert scale, and a per-case question budget. MediQ requires hand-curated multiple-choice options for each case and a single 100-patient pool is impractical for it at this option granularity, so we evaluate every benchmark column on disjoint samples and report each separately; for fairness, every other baseline is evaluated on the same samples. We report two scoring conventions:
  - *force-final* (paper-faithful): cases that exhaust the budget without crossing the Likert threshold are forced to commit on the highest-confidence intermediate choice; coverage is 100% by construction.
  - *strict-abstain*: those budget-exhausted cases are recorded as abstentions instead. We report this as a paper-side ablation because force-final inflates coverage to 100% and counts the model's unconfident commits as commits, depressing selective accuracy;

strict-abstain isolates what MediQ would do if its own threshold-based abstention rule were respected at scoring time.

- **MEDDxAgent (paper-best) (14).** The recommended best iterative configuration: a single-LLM diagnosis agent with dynamic 5-shot fewshot retrieval (active case excluded), top-2 PubMed retrieval, and a fixed schedule of 3 iterations $\times$ 5 history-taking questions per iteration. MEDDxAgent has no abstention mechanism by design, so its coverage is 100% by construction.
- **MEDDxAgent (closed-world variant).** Same orchestration as paper-best but with PubMed retrieval and dynamic fewshot disabled, isolating the contribution of MEDDxAgent's external retrieval scaffolding from the rest of its orchestration.
- **AgentClinic (40).** AgentClinic's bundled LLM doctor: short single-turn questions, optional *request test* actions (resolved as "I don't know" on the DDxPlus columns where no measurement agent is available), and commitment within the published question budget.
- **CoD / DiagnosisGPT-34B (31).** The released DiagnosisGPT-34B checkpoint (Yi-34B fine-tune) and its broad-domain disease retriever, run inside the published CoD dialogue loop with the default per-case inquiry budget and the built-in commit confidence threshold. CoD's free-text final diagnoses are mapped to the active closed disease universe through a one-shot LLM-judge prompt that compares the free-form output (and the model's top-confidence candidates) against the closed list and either picks the closest semantic match or returns `Unknown`; substring matching is brittle for CoD's verbose differentials, so this LLM mapping is necessary to score CoD on the same closed universe as the other baselines.
- **UoT (28).** LLM information-pursuit method that expands a yes/no question tree and selects the question maximising expected information gain estimated from the LLM's own answer probabilities. UoT's tree expansion stalls when the patient cannot answer cleanly: hedged or "I don't know" responses are correctly parsed as "no information" by our yes/no parser, but the resulting empty turn provides no posterior update so the tree cannot make progress. The effect is most pronounced on the HT-sim column, where the vendor patient prompt explicitly defaults to "I don't know" for any feature outside the profile, which substantially deflates UoT's reported numbers there.

AgentClinic protocol adaptations. The AgentClinic-MedQA column reuses AgentClinic's published patient and moderator agents unchanged. Two evaluation-level adjustments are needed for cross-baseline comparability: (i) the AC-50 closed disease universe is pinned and every doctor must commit within or abstain over this set (patient utterances are unaffected); (ii) every doctor opens with the same fixed greeting so the chief complaint is elicited from the patient simulator on the first turn rather than implicitly assumed.

C.15 True diagnosis absent from the KB

For 50 DDxPlus cases, with gpt-5.4-nano as sensor and patient, we remove the true disease from the priors, likelihoods, and EIG candidates. Every commit is then wrong, so abstention is the only correct outcome. The standalone doctor receives the same 48 disease names and an instruction to abstain if none matches. At τ=0.9, MOBAYES abstains on 88% of cases and the standalone on 46% (Table 22). The standalone commits despite the instruction and once names the excluded disease. MOBAYES commits less often at every threshold, and the threshold sets this trade-off explicitly.

C.16 Sensitivity to KB noise and coverage

We replay the logged evidence of 100 PatientSim dialogues through corrupted copies of the empirical DDxPlus KB. Question selection and extraction stay frozen, so the injected noise is the only source of error. We add log-normal noise of scale σ to each likelihood, or blank a fraction ρ of entries. Each level averages 20 draws at a fixed τ=0.3. Replaying the clean KB reproduces 97 of 100 logged diagnoses. DHS stays flat up to σ=0.5 but drops sharply once a large share of entries is blank (Table 23). The gains thus tolerate imprecise likelihoods but require the KB to cover the feature space.

Table 22: True disease removed from the candidate space (n=50). Top, outcomes at τ=0.9. Bottom, commit rate by threshold.

Outcome	MoBayes	Standalone
Abstained	**44 (88%)**	23 (46%)
Wrong in-list commit	6 (12%)	26 (52%)
Names excluded disease	0	1 (2%)

τ	0.5	0.6	0.7	0.8	0.9	0.95
MoBayes	36%	28%	26%	22%	12%	12%
Standalone	54%	54%	54%	54%	54%	54%

Table 23: DHS under likelihood noise σ and KB blanking ρ, mean $\pm$ s.d. over 20 draws (n=100, τ=0.3).

σ	0	0.1	0.25	0.5	1.0	2.0
DHS	88.0	87.8±0.7	87.7±1.2	87.6±1.5	84.5±1.9	69.0±3.0
ρ	0	0.1	0.25	0.5	0.75	1.0
DHS	88.0	85.9±1.4	82.1±1.9	70.0±3.0	37.1±5.0	0.0

C.17 Parser audit

We hand-label 200 sampled patient messages and validate the labels with a blinded LLM judge. The parser extracts the correct value in 92.5% of messages. Most errors concern the confidence label, and polarity flips occur in only 1% of messages. The judge agrees with the human labels on 96.5% of values and 91.5% of confidence labels. An automatic check against the patient profiles gives 98.7% agreement on binary findings. Residual errors enter the engine as bounded soft evidence (Appendix A.1).

C.18 Model-matched sensor and standalone pairings

Table 2 pairs each MoBayes sensor with a larger standalone sibling. Table 24 completes this grid for two model families. In both families, the small sensor with MoBayes outperforms the large standalone doctor. The gap between small and large sensors within MoBayes is much smaller.

Table 24: Model-matched pairings on DDxPlus PatientSim (DHS). MoBayes rows commit on every case; standalone doctors at their natural operating point.

Configuration	GPT-5.4	Gemini 3.1
Small standalone	50.9	69.6
Large standalone	69.4	72.6
Small sensor + MoBayes	83.7	86.4
Large sensor + MoBayes	85.1	88.9

C.19 Full AgentClinic-MedQA

The sampled AgentClinic-MedQA columns cover 31 diseases (Appendix C.14). For the full set of 213 cases and 177 diseases, we build the KB following the pipeline of Appendix B.4, with adaptations, giving 828 categorical features. AgentClinic's patient agent simulates the patient, and gpt-5.4-nano serves as both sensor and patient model. On the full set, MoBayes reaches Top-1 75.1 [68.8, 81.0] and DHS 86.2 [82.1, 89.8], with bootstrap 95% CIs and τ^* re-tuned per resample. This result falls inside the confidence intervals of both sampled columns, so the case selection does not favour MoBayes.

C.20 Diagnosis under a staged examination protocol

In clinical practice, questions follow a protocol with fixed categories and a fixed order. VivaBench (48) is one such protocol, with history, examination, investigation, and imaging stages in sequence. We

run a feasibility study on it. We take 50 Cardiovascular and Metabolic cases and a 29-disease universe consolidated by an LLM, and elicit a 264-feature KB for these cases. Each feature is labelled with its stage, and laboratory or imaging results are ordinary features with disease-conditional likelihoods. Within each stage, EIG selects among the questions labelled with that stage, under a per-stage budget. Once a stage ends, its questions are no longer available, and the posterior carries over to the next stage. Patient answers are looked up from a table of findings that an LLM extracted once from each case. MOBAYES reaches Top-1 44 [30, 58]. This is a feasibility check on a 50-case, single-specialty subset, so we make no comparative claim.

D Extended related work

This appendix expands on Section 2 with a system-by-system discussion of the four families that motivate MOBAYES's strict separation, ending with a synthesis of where each family places the LLM relative to the diagnostic decision loop. Table 25 summarises which of the three core diagnostic components (knowledge base, posterior tracking, question selection) each system delegates to the LLM versus to an external module.

Auditable probabilistic CDSS. The classical CDSS literature explored multiple probabilistic formalisms for diagnosis. MYCIN (1) represented diagnostic uncertainty through certainty factors and rule-based inference and was evaluated against infectious-disease specialists in a blinded protocol (17). de Dombal et al. (2) showed that even a flat naïve Bayes model could surpass senior surgeons on acute abdominal pain when fed structured signs and symptoms. INTERNIST-I and its successor QMR (3) scaled heuristic disease-profile scoring to several hundred internal-medicine conditions, and DXplain (4) maintained a disease–finding matrix with modified Bayesian updates over thousands of diseases. Later work introduced richer graphical-model representations: Pathfinder (5) placed lymph-node pathology under a full Bayesian belief network, and CASNET (18) expressed glaucoma management through a layered causal model linking findings to disease states to therapeutic actions. All of these systems shared the same structural pattern, an explicit probabilistic engine over a curated knowledge base coupled to a structured-input modality that demanded the clinician encode the case in the system's vocabulary; their failure to scale into routine practice, captured by the "Greek Oracle" diagnosis (19), was an interface failure rather than a reasoning failure.

End-to-end LLM-based diagnostic systems. Modern LLM-driven systems span single-turn QA, multi-turn dialogue, and domain-specialised model checkpoints. Med-PaLM 2 (6) and Med-Gemini (7) pushed standalone-LLM medical question-answering to expert level on MedQA-style benchmarks but operate single-turn with no belief state and no information-seeking behaviour. AMIE (8) extended this paradigm to multi-turn dialogue through self-play training between simulated doctor and patient roles, achieving expert-comparable diagnostic performance on a curated history-taking benchmark. Xu et al. (20) aligned multi-turn medical dialogue with abductive clinical reasoning, while DocCHA (15) decomposes the diagnostic process into modular sub-tasks (symptom collection, history acquisition, causal-graph construction) but routes all decisions through the LLM. A parallel line of domain-specialised checkpoints fine-tunes base LLMs on clinical corpora, exposing only an end-to-end interface. Fully Open Meditron (49) addresses training-pipeline transparency through clinician-audited data and a reproducible data construction and training framework. Across all of these, no calibrated posterior, principled stopping rule, or deployment-time abstention mechanism is exposed: language and reasoning remain entangled inside a single generative model.

Probability–language hybrids. Hybrid systems inject probabilistic structure into LLM-driven pipelines while keeping the LLM in at least one part of the inference loop. BED-LLM (9) formulates each turn as Bayesian Experimental Design: a Rao-Blackwellised estimator extracts a posterior over a target variable from the LLM's own predictive distributions and selects the question that maximises expected information gain, with the probabilistic model itself sourced from the language model. ACTMED (26) applies Bayesian experimental design to clinical test selection. The LLM simulates patient states and supports the belief update, so it again supplies the probabilities. MOBAYES instead reads them from an explicit KB. DeLLMa (10) pairs an LLM with classical decision theory, eliciting a utility function from the model and selecting actions that maximise expected utility. BIRD (11) aligns a Bayesian network with LLM abductions and then performs a deductive inference step, with the goal of producing controllable probability estimates more reliable than direct LLM scoring. APP (12)

combines clinical guidelines with entropy-based active learning and Bayesian-style updates for human-centric medical dialogue. Sun et al. (27) couple an LLM parser with a separately RL-trained planner. UoT (28) performs information-pursuit through an LLM-generated binary question tree with Q-value-based selection, originally for the 20-Questions setting. MediQ (13) contributes the abstention mechanism most directly relevant to clinical dialogue: it shows that pairing question-asking with explicit scale-based abstention recovers the accuracy that naive prompting otherwise loses. CoD / DiagnosisGPT (31) fine-tunes a clinical LLM with an explicit confidence head and a $9{,}604$-disease retriever, while Kim et al. (16) examine the privacy and safety trade-offs of clinical fine-tuning more broadly. Across the family, the LLM remains responsible for at least one of: producing the probabilistic distribution, generating candidate questions, or scoring outputs, which keeps language and probability entangled.

Agentic diagnostic frameworks. A complementary family wraps LLMs in multi-agent orchestration. MEDDxAgent (14) runs a four-agent loop (DDx-driver, history-taking, knowledge-retrieval, diagnosis) with PubMed or closed-world retrieval and optional dynamic few-shot prompting, where every agent calls the same backbone LLM. AgentClinic (40) simulates a clinical environment through doctor, patient, measurement, and moderator agents, each implemented as an LLM with role-specific prompts. Li et al. (13) package their abstention-aware pipeline as a five-step Expert framework (Assessment, Abstention, Question, Integration, Decision). These systems modularise the workflow but rely on heuristic role-playing rather than information-theoretic objectives for question selection, and the LLM is still the primary inference engine inside each role; we adapt several of them as baselines (Section 4.3, Appendix C.14).

Safety, calibration, and bias risks. The structural absence of a calibrated probabilistic state in end-to-end LLMs has been linked to several measurable safety risks. Savage et al. (21) report that verbalised confidence from clinical LLMs shows minimal variation between correct and incorrect diagnoses, undermining its use as an abstention signal. Omar et al. (23) demonstrate that frontier clinical LLMs are highly vulnerable to adversarial hallucination during clinical decision support, with multi-model assurance providing only partial mitigation. The medical knowledge encoded in these models is also skewed: Omiye et al. (24) document propagation of race-based clinical patterns, and Nimo et al. (25) report substantial accuracy gaps on African medical questions relative to Western populations. Recent benchmarks such as CLIMB (50) expose limitations in LLMs' ability to track multiple diagnostic hypotheses during dialogue. We hypothesise that keeping diagnostic state implicit in language generation contributes to this failure, and that a Bayesian harness with explicit belief tracking, such as MOBAYES, could help address it.

Why MOBAYES differs. Across the four families above, no system keeps the LLM strictly out of the diagnostic decision loop. Classical CDSS provided the auditable statistical engine but coupled it to a structured-input interface that excluded clinicians from natural conversation. End-to-end LLMs solved the interface but folded conversation, question selection, diagnosis, and stopping into a single generative model. Hybrid systems and agentic frameworks reintroduce probabilistic components and modular workflows but still rely on the LLM as a probability source, candidate generator, planner substrate, or reasoning state. MOBAYES's contribution is not a new probabilistic engine and not a new LLM, but a deployment pattern that reuses classical statistical reasoning and modern conversational interfaces without letting either contaminate the other. Table 25 makes this explicit: for each surveyed system, it indicates whether the LLM is responsible for the knowledge base, the posterior, the question-selection criterion, and the abstention rule.

Table 25: System-by-system comparison of diagnostic dialogue approaches against the architectural axes our position depends on (Section 2). **MT**: supports multi-turn dialogue. **Belief**: explicit probabilistic posterior over diseases (✓), an LLM-derived approximate posterior (Approx), or none (×). **KB**: clinical knowledge source (Cur. = curated; Trained = baked into model weights; RAG = retrieval over external corpus; LLM-el. = elicited from a frontier LLM; "–" = none). **Q-sel.**: question-selection criterion (LLM = LLM-internal heuristic; EIG = expected information gain; Tree = lookahead tree search; RL = learned policy; Conf. = conformal-set shrinkage; Util. = expected-utility maximisation; "–" = no question-asking). **Abst.**: abstention mechanism (Heur. = ad hoc; Calib. = calibrated; Ctrl. = continuously controllable; "–" = none). **LLM-free**: the LLM is not in the diagnostic decision loop. Systems marked with $*$ have no public code release or are closed products and are therefore cited but not reproduced as baselines in our experiments.

System	MT	Belief	KB	Q-sel.	Abst.	LLM-free
Classical CDSS						
MYCIN* (1)	✗	✓	Cur.	–	–	✓
INTERNIST/QMR* (3)	✗	✓	Cur.	–	–	✓
DXplain* (4)	✗	✓	Cur.	–	–	✓
Pathfinder* (5)	✗	✓	Cur.	–	–	✓
CASNET* (18)	✗	✓	Cur.	–	–	✓
End-to-end frontier LLMs						
Med-PaLM 2* (6)	✗	✗	Trained	–	–	✗
Med-Gemini* (7)	✗	✗	Trained	–	–	✗
AMIE* (8)	✓	✗	Trained	LLM	–	✗
DocCHA* (15)	✓	✗	Cur.	LLM	Heur.	✗
Domain-tuned medical LLMs						
Meditron (51)	✓	✗	Trained	–	–	✗
MedGemma (52)	✓	✗	Trained	–	–	✗
HuatuoGPT-o1 (53)	✓	✗	Trained	–	–	✗
CoD / DiagnosisGPT (31)	✓	Approx	Trained	Entropy	Heur.	✗
DoctorAgent-RL (54)	✓	✗	Trained	RL	–	✗
Probability–language hybrids						
BED-LLM* (9)	✓	Approx	–	EIG	–	✗
APP* (12)	✓	Approx	RAG	Entropy	–	✗
Planner-LLM (27)	✓	✗	–	RL	–	✗
UoT (28)	✓	✗	–	Tree+EIG	–	✗
CIP* (55)	✓	✗	–	Conf.	Calib.	✗
MedKGI* (56)	✓	Approx	Cur.	EIG	Heur.	✗
MedClarify* (57)	✓	Approx	–	EIG	–	✗
Agentic frameworks						
MEDDxAgent (14)	✓	✗	RAG	LLM	–	✗
AgentClinic (40)	✓	✗	–	LLM	–	✗
MediQ (13)	✓	✗	–	LLM	Heur.	✗
MoBayes (Ours)	✓	✓	Cur./LLM-el.	EIG	Ctrl.	✓

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the architectural-separation thesis (LLM as language interface, deterministic Bayesian engine for reasoning), the controllable abstention mechanism, and the statistical separation gap between same-family standalone doctors and MoBayes sensors. Each claim is evaluated in Section 5 against Tables 2 and 1, and discussed in Section 6.

Guidelines:

- The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 contains an explicit Limitations paragraph covering (i) the requirement for an explicit knowledge base (LLM elicitation removes the dependence on labelled clinical data, but the resulting KB inherits the underlying LLM's noise so the engine is at most as robust as that LLM), (ii) reliance on simulated rather than real patients, and (iii) inability to integrate volunteered evidence outside the asked question.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The soft-evidence update is derived in Appendix A.1, the EIG criterion in Appendix A.2, and the question-budget analysis in Appendix A.3; assumptions are stated at the point of derivation.

   Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The diagnostic-loop pseudocode is in Appendix B.1; all prompts (parsing, verbaliser, bulk intake, patient simulator, standalone doctor, disease matching) are reproduced verbatim in Appendix B.2; knowledge base construction in Appendix B.4; per-method baseline configurations and protocol adaptations in Appendix C.14; patient-simulator adaptations across the three columns in Appendix B.5. Code is available at `https://github.com/EPFLiGHT/MoBayes`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The public repository (`https://github.com/EPFLiGHT/MoBayes`) contains the bench framework, all baseline adapters, the unified LLM-judge for closed-set normalisation, the LLM-elicited knowledge bases we produced, and the prompt templates of Appendix B.2.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Datasets, baselines, simulators, and metrics are described in Section 4.1–4.4; per-method configurations and the threshold-tuning protocol are in Appendix C.14 and Appendix C.3.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Appendix C.13 reports 95% percentile bootstrap confidence intervals (B=2000) for every cell of Tables 2 and 21.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Per-token API pricing and per-patient cost are reported in Appendix C.7; the only fine-tuned reference (DiagnosisGPT-34B) is run as a single Slurm batch on one A100-80GB node, all other methods run via API without local GPU.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work conforms with the NeurIPS Code of Ethics. No human subjects are involved and no patient data is fine-tuned into model weights.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section 6 contains an explicit Broader Impact paragraph.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The released artifacts contain no patient data and no model weights. The LLM-elicited knowledge bases are synthetic medical-reference tables analogous to a textbook differential, the curated samples are subsets of already-released datasets, and the bench framework wraps third-party LLM APIs without redistributing them. No high-misuse-risk model or scraped dataset is released by us.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All third-party assets are cited at first use: DDxPlus (32), AgentClinic and the AgentClinic-MedQA scenarios (40), MEDDxAgent's history-taking simulator and StreamBench loader (14), and PatientSim's persona taxonomy (43). We use each asset within the bounds of its stated license (publicly available at each project's repository) and credit the original creators in the relevant sections of the paper. Frontier LLMs are accessed through the OpenAI, Google, and OpenRouter APIs under their respective terms of service.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code repository ships the LLM-elicited knowledge bases and the bench framework, each with a README describing scope, construction, and usage.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: All patients in our experiments are LLM-driven simulators. No human subjects, crowdworkers, or annotators were involved in the evaluation.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: No human subjects were involved in this work (see Q14); IRB approval is therefore not applicable. All patient profiles are derived from publicly available datasets or generated by frontier LLMs.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The role of LLMs in the proposed method (language interface and knowledge-base elicitor) is described in Section 3.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.