Natural Language Grounded Reinforcement Learning for Clinical Decision-Making in Virtual Patient Simulations

Niyel Hassan
High Technology High School

Benjamin Liu
Stanford University

Jason Tsai
Stanford University

Jeffrey Jopling
Johns Hopkins University

Dana Lin
Stanford University

Edward Melcer
Carleton University

Cara Liebert
Stanford University

Abstract

We present a reinforcement learning framework for training agents in simulated clinical diagnostic tasks within virtual patient simulations. Each patient case is cast as a Markov Decision Process with a hybrid state that fuses semantic encodings of clinical text with structured physiology and a masked Proximal Policy Optimization policy that enforces clinical action feasibility. The learned policy is stable and competent, achieving a recall of 0.75 for clinically indicated actions while avoiding over 96% of harmful actions. Domain-specific language encoders materially improve performance, underscoring the value of a language-grounded state. Crucially, we find that a conservative checklist strategy, which favors completeness over efficiency, reveals disparities across specialties and demographics, including a safety drop in geriatric cases. Our framework offers a rigorous testbed and strong baseline for language-based clinical policy learning and clarifies targets for improving efficiency, generalization, and fairness in reinforcement learning agents for clinical decision-making.

1 Introduction

Competency-based medical education aims to ensure trainees can perform the professional tasks required for independent practice at a consistent level of safety and quality [32]. Entrustable Professional Activities (EPAs) map broad competencies to concrete, observable units of clinical work that can be entrusted once competence is demonstrated [32, 30]. Defensible entrustment hinges on reliable assessment of clinical reasoning across time and contexts—a process that is difficult to scale via direct observation and burdensome for faculty, motivating the use of virtual patient simulations as standardized, lower-friction settings to practice and assess sequential diagnostic decision-making; in these environments clinical reasoning unfolds as sequences of information-gathering and interventions under uncertainty, lending itself to computational formalization [16, 13, 35].

Reinforcement learning (RL) provides a principled framework for learning decision policies from interaction, optimizing long-horizon objectives by balancing exploration and exploitation [35, 11]. Applying RL to clinical simulations highlights long-standing design challenges—representing the clinical state so salient semantics are preserved, constraining actions to feasible and clinically appropriate choices, and specifying rewards that reflect safety, efficiency, and diagnostic value [35, 13]. State representation is particularly consequential because, while structured physiological variables are informative, much clinical context—presenting complaints, history, and textual test results—is inherently linguistic; representing environment state with natural language can therefore improve sample efficiency, robustness, generalization, and interpretability [18, 21, 8, 29].

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: The Second Workshop on GenAI for Health: Potential, Trust, and Policy Compliance.

At the same time, evaluating clinical reasoning with modern language models has outgrown reliance on static, multiple-choice benchmarks. High scores on such tests do not necessarily translate into reliable, context-sensitive decision-making in interactive settings [3, 4]. Recent evaluation paradigms therefore emphasize multi-turn interaction, tool use, and sequential information-gathering within simulated clinical workflows [26]. EPA-style simulations provide a complementary testbed in this spirit: they offer standardized cases with explicit action constraints and clinically grounded utility signals, while retaining the sequential structure of real diagnostic work.

This paper situates EPA simulations within an RL formulation that leverages natural language as a carrier of clinical state. Concretely, we model diagnostic encounters as a finite-horizon decision process in which the state integrates semantic embeddings of the initial presentation and accumulated findings with structured physiology. Actions are drawn from a constrained, case-specific set provided by the simulation. In our experiments, the dataset is based on the *ENTRUST* virtual patient simulation platform [16], which teaches surgical decision-making through comprehensive case vignettes. Each case includes a diverse set of actions—imaging, laboratory tests, and physical examinations—with utilities spanning indicated, harmful, and neutral options. Reward signals reflect stepwise costs and clinically derived utilities [2]. Within this setting, we develop a policy optimized with an on-policy algorithm that respects dynamic action masks for clinical feasibility [27, 20].

Beyond aggregate performance, our evaluation framework is designed to probe properties of practical importance in medical education and safety. We assess diagnostic completeness and parsimony using standard information-retrieval metrics; examine generalization by stratifying performance across clinical specialties and patient demographics; and analyze qualitative trajectories to understand common success and failure modes. Finally, because collecting the "right" information is only useful insofar as it supports downstream clinical reasoning, we include a probe that summarizes agent trajectories for use in a separate question-answering task, thereby testing whether gathered evidence is organized in a way that aids external decision support [3, 4, 26].

Taken together, this language-grounded RL perspective on surgical EPAs aims to connect scalable, simulation-based assessment with policy learning that is explicit about clinical constraints and utilities. By centering representation, feasibility, and evaluation on clinically meaningful constructs, the framework is intended to support rigorous study of how agents acquire diagnostic strategies—and where they fall short—in settings that matter for entrustment and training [32, 16].

2 Related Work

Reinforcement Learning in Healthcare RL is a natural fit for sequential clinical decisions and has been studied for dynamic treatment regimens and critical care management [35, 11, 1]. Success in this setting depends on careful MDP design: states must capture clinically salient context, actions must reflect feasible interventions, and rewards must align with safety and diagnostic value [35, 13]. We focus on the state representation problem as a lever for reliable policy learning within constrained clinical simulations [18].

Natural Language as State Representation Text encodings have progressed from hand-engineered features to learned representations that capture semantics from interaction. Early work in text-based environments showed gains from sequence models and separate state/action embeddings [21, 8]. Subsequent studies indicate that natural-language state descriptions can improve robustness and convergence—even when other modalities are available—while also offering interpretability [29]. Our formulation adopts this paradigm by encoding clinical presentations and accumulated findings using domain-specific biomedical language models [2].

Evaluating LLMs in Clinical Environments High performance on multiple-choice medical QA (e.g., MedQA, MedMCQA) does not fully capture interactive, uncertainty-laden clinical reasoning [12, 22]. Recent benchmarks therefore emphasize multi-turn interaction, tool use, and sequential information-gathering [3, 4, 26]. EPA-style simulations align with this shift since they provide standardized cases, explicit action constraints, and clinically grounded utilities. Our study builds on this work by casting EPA simulations as an RL environment and studying how language-grounded policies acquire diagnostic strategies through interaction [26].

3 Method

Our approach models the sequential clinical decision-making task within a reinforcement learning (RL) framework. We formalize the problem as a Markov Decision Process (MDP), where the agent's policy is optimized using a state-of-the-art on-policy algorithm. The state representation is engineered to integrate semantic information from clinical text with structured physiological data, enabling the agent to navigate complex diagnostic scenarios.

3.1 Problem Formulation as a Markov Decision Process

We define the diagnostic task as a finite-horizon, discounted MDP, represented by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$. The agent's goal is to learn a policy $\pi(a|s)$ that maximizes the expected cumulative discounted reward, $G_t = \sum_{k=0}^{T-t-1} \gamma^k r_{t+k}$, where T is the episode horizon and $\gamma \in [0, 1]$ is the discount factor.

3.1.1 State Space ($\mathcal{S}$)

The state $s_t \in \mathcal{S}$ at timestep t is a fixed-dimension vector constructed to provide a comprehensive summary of the clinical encounter. It concatenates four distinct components, $s_t = [e_{\text{init}} \parallel e_{\text{hist},t} \parallel v_{\text{phys},t} \parallel \tau_t]$, where $\parallel$ denotes concatenation.

Initial Case Embedding (e_{init}). A static, d_{emb}-dimensional vector representing the initial patient presentation. This embedding is generated by encoding the concatenation of the patient's history, chief complaint, and initial physical exam findings using a pre-trained biomedical language model, Bio_ClinicalBERT [2]. We compute the final embedding via mean-pooling of the last hidden states of all input tokens and apply L2 normalization. This static component provides a constant contextual anchor throughout the episode.

Historical Action Embedding ($e_{\text{hist},t}$). A dynamic, d_{emb}-dimensional vector that summarizes the semantic content of all information gathered up to timestep t. The result of each action a_i (e.g., lab result text) is encoded into an embedding $e_{\text{res},i}$ using the same language model. The historical embedding is the L2-normalized running average of these result embeddings: $e_{\text{hist},t} = \text{L2Norm}(\frac{1}{t}\sum_{i=1}^{t-1} e_{\text{res},i})$. This provides an evolving summary of the diagnostic findings [10, 19].

Physiological State Vector ($v_{\text{phys},t}$). A d_{lab}-dimensional numerical vector representing the patient's known physiological parameters, including initial vital signs and any laboratory values revealed by previous actions. To ensure a consistent scale across different measures, each value is z-score normalized using the mean and standard deviation computed from the entire training portion of the dataset.

Time Step Feature (τ_t). A scalar value $\tau_t = t/T_{\max}$ representing the normalized progression of the episode, where $T_{\max}$ is the maximum allowed number of steps. This feature allows the policy to be time-aware.

The resulting state vector has a total dimension of $2d_{\text{emb}} + d_{\text{lab}} + 1$.

3.1.2 Action Space ($\mathcal{A}$)

The action space $\mathcal{A}$ is a discrete set of all unique diagnostic and therapeutic actions available across all cases in the dataset. For each specific case c, a binary action mask $M_c \in \{0, 1\}^{|\mathcal{A}|}$ is provided by the environment. This mask restricts the agent to a subset of clinically relevant actions and is updated at each step to prevent the re-selection of previously taken actions, ensuring a realistic and constrained decision space.

3.1.3 Reward Function (R)

The reward function is engineered to guide the agent toward policies that are both diagnostically accurate and efficient. The reward r_t received at timestep t after taking action a_t is a sum of three components:

$$R(s_t, a_t) = r_{\text{step}} + r_{\text{action}}(a_t) + r_{\text{terminal}}(s_{t+1}) \tag{1}$$

- **Step Penalty (r_{step}):** A small negative constant ($r_{\text{step}} = -0.2$) is applied at every timestep. This incentivizes the agent to solve the case in as few steps as possible, promoting efficiency.
- **Action Score (r_{action}):** Each action a_t has a pre-defined clinical utility score, $S(a_t)$, provided by the simulation environment. This score is given as an immediate reward, $r_{\text{action}}(a_t) = S(a_t)/100$, directly rewarding clinically valuable actions and penalizing detrimental ones.
- **Terminal Reward (r_{terminal}):** A large bonus or penalty is awarded only at the end of an episode. An episode terminates if all designated positive-utility actions for the case have been selected (solved), or if the step limit $T_{\max}$ is reached (unsolved).
  - If solved at step $T < T_{\max}$: A large positive reward is given, scaled by the remaining time to encourage speed: $r_{\text{terminal}} = R_{\text{solve}} + R_{\text{speed}} \cdot (1 - T/T_{\max})$, where we set $R_{\text{solve}} = 10$ and $R_{\text{speed}} = 5$.
  - If unsolved at step $T_{\max}$: A large negative penalty is applied, scaled by the fraction of missed positive-utility actions: $r_{\text{terminal}} = -R_{\text{fail}} \cdot (1 - \text{Recall})$, where $R_{\text{fail}} = 10$.

3.2 Policy Optimization with Masked PPO

We train the agent using Proximal Policy Optimization (PPO) [27], an on-policy actor-critic algorithm known for its sample efficiency and stable training dynamics. To handle the constrained action space, we employ a variant that incorporates action masking directly into the policy distribution.

3.2.1 Agent Architecture

The agent utilizes a shared-parameter actor-critic architecture [20] with a two-layer Multi-Layer Perceptron (MLP) backbone. The network takes the state vector $s_t \in \mathbb{R}^{2d_{\text{emb}}+d_{\text{lab}}+1}$ as input. The shared backbone consists of two hidden layers of 64 units each, with $\tanh$ activation functions. Network weights are initialized using orthogonal initialization, which has been shown to improve stability in deep RL settings [9]. The backbone outputs a shared feature representation that feeds into two separate linear heads:

1. The **Actor Head** outputs logits $\mathbf{l} \in \mathbb{R}^{|\mathcal{A}|}$ over the entire action space. The action mask M_c is applied by setting the logits of invalid actions to negative infinity ($-\infty$) before the softmax operation, ensuring that the probability of selecting an invalid action is zero. The final stochastic policy is given by $\pi_\theta(a \mid s_t) = \text{Softmax}\big(\mathbf{l} - \infty \cdot (1 - M_c)\big)_a$.
2. The **Critic Head** outputs a single scalar value $V_\phi(s_t)$, which estimates the expected cumulative return (the state-value) from state s_t.

3.2.2 Training Procedure

The actor and critic networks are optimized jointly. We collect trajectories using multiple parallel environments and compute advantage estimates using Generalized Advantage Estimation (GAE) [28] to reduce variance. The composite loss function is:

$$L(\theta, \phi) = \mathbb{E}_t \left[-L^{\text{CLIP}}(\theta) + c_1 L^{\text{VF}}(\phi) - c_2 S[\pi_\theta](s_t) \right] \tag{2}$$

where L^{CLIP} is the PPO clipped surrogate objective, L^{VF} is the squared-error value function loss, and $S[\pi_\theta]$ is an entropy bonus to encourage exploration. c_1 and c_2 are weighting coefficients. The clipped objective is:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(\rho_t(\theta)\hat{A}_t, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{3}$$

where $\rho_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio, $\hat{A}_t$ is the GAE advantage estimate, and ϵ is the clipping hyperparameter. The training procedure is detailed in Algorithm 1. We use the Adam optimizer [14] for gradient-based updates. Full implementation details and hyperparameter settings are provided in Appendix D.

3.3 Experimental Setup and Evaluation

We evaluate the agent's performance on a held-out set of medical cases, using metrics that capture diagnostic accuracy, efficiency, and the clinical coherence of its learned behavior.

Algorithm 1 Masked PPO for Clinical Decision-Making

Input: Hyperparameters: num. envs N_{env}, rollout length $T_{rollout}$, epochs K, minibatch size M, learning rate α, γ, GAE-λ, clip ϵ, entropy coef. c_2.
Initialize actor network π_θ and critic network V_ϕ.
Initialize N_{env} parallel environments.
for iteration $= 1, 2, \ldots, N_{iter}$ **do**
    Initialize a trajectory storage buffer $\mathcal{D}$.
    Reset environments and get initial observations $\mathbf{s}_0$ and masks $\mathbf{M}_0$.
    for step $t = 0, \ldots, T_{rollout} - 1$ **do**
        With probability from $\pi_\theta(\cdot|\mathbf{s}_t, \mathbf{M}_t)$, sample actions $\mathbf{a}_t$.
        Compute action log-probabilities $\log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t, \mathbf{M}_t)$ and values $v_t = V_\phi(\mathbf{s}_t)$.
        Execute actions $\mathbf{a}_t$, receive rewards $\mathbf{r}_t$, next states $\mathbf{s}_{t+1}$, done $\mathbf{d}_t$, and next masks $\mathbf{M}_{t+1}$.
        Store $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{r}_t, \mathbf{d}_t, \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t, \mathbf{M}_t), v_t)$ in $\mathcal{D}$.
        $\mathbf{s}_t \leftarrow \mathbf{s}_{t+1}, \mathbf{M}_t \leftarrow \mathbf{M}_{t+1}$.
    end for
    Compute last value $v_{last} = V_\phi(\mathbf{s}_{T_{rollout}})$.
    Compute advantage estimates $\hat{\mathbf{A}}_t$ and returns $\mathbf{R}_t$ for all timesteps in $\mathcal{D}$ using GAE.
    for epoch $= 1, \ldots, K$ **do**
        Shuffle transitions in $\mathcal{D}$.
        for each minibatch of size M from $\mathcal{D}$ **do**
            Evaluate current policy: new log-probs, values V_ϕ, and entropy S.
            Compute policy ratio $\rho_t(\theta) = \exp(\text{new\_log\_probs} - \text{old\_log\_probs})$.
            Compute clipped surrogate objective L^{CLIP} via Eq. 3.
            Compute value loss $L^{\text{VF}} = (\mathbf{R}_t - V_\phi)^2$.
            Compute total loss $L = -L^{\text{CLIP}} + c_1 L^{\text{VF}} - c_2 S$.
            Update parameters θ, ϕ using Adam: $(\theta, \phi) \leftarrow (\theta, \phi) - \alpha \nabla_{(\theta,\phi)} L$.
        end for
    end for
end for

Performance Metrics. A complete description of the dataset and environment is available in Appendix C. We assess the agent's policy on the test set using a deterministic protocol, evaluating both overall performance and diagnostic accuracy. Overall performance is measured by the average number of steps per episode and case completion rate. Diagnostic accuracy is evaluated by comparing the agent's selected actions, $\mathcal{A}_{\text{taken}}$, against the set of required positive-utility actions, $\mathcal{A}_c^+$, and negative-utility actions, $\mathcal{A}_c^-$, for each case c. Key metrics include **recall** ($|\mathcal{A}_{\text{taken}} \cap \mathcal{A}_c^+|/|\mathcal{A}_c^+|$), **precision** ($|\mathcal{A}_{\text{taken}} \cap \mathcal{A}_c^+|/|\mathcal{A}_{\text{taken}}|$), their harmonic mean (**F1 score**), and **specificity** (the fraction of negative-utility actions correctly avoided). To gauge the clinical utility of the agent's information-gathering, we perform a downstream question-answering (QA) task, detailed in Appendix B. A summary of the agent's trajectory serves as context for an external LLM (Gemma 3 27B-IT [31]) to answer case-specific multiple-choice questions.

4 Results

We evaluate our RL agent's performance through its training progression and a series of ablation studies on a held-out test set of clinical cases. We assess overall performance, diagnostic accuracy, and efficiency to validate our method and understand the contributions of its core components.

4.1 Overall Agent Performance

The agent successfully learns a stable policy, demonstrated by the monotonic increase in cumulative reward over 100,000 training episodes (Figure 1a). The policy optimizes for diagnostic comprehensiveness at the cost of efficiency, a trade-off revealed by the final performance metrics. The agent rapidly learns to identify the majority of necessary clinical actions (recall = 0.75) and consistently avoids harmful ones (specificity > 0.96), as shown in Figures 1b and 1c. However, its lower precision (0.55) indicates a tendency to select superfluous, diagnostically neutral actions, resulting in a final F1 score of 0.60. This suggests the agent adopts a safe but exhaustive information-gathering strategy.

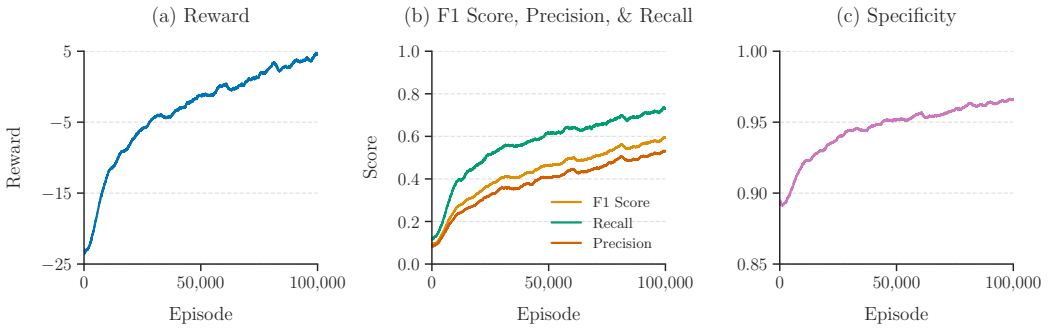

Figure 1: Agent training performance over 100,000 episodes.

4.2 Performance Analysis Across Subgroups

Table 1: Performance breakdown by medical specialty, averaged over 3 seeds.

Specialty	Solved (%)	F1 Score	Recall	Precision	Specificity
Plastic and Maxillofacial	28.6	0.610	0.696	0.570	0.975
Vascular Surgery	16.0	0.582	0.625	0.593	0.981
General Surgery	13.8	0.519	0.653	0.474	0.960
Pediatric Surgery	6.3	0.513	0.653	0.489	0.972
Cardiothoracic Surgery	14.3	0.437	0.546	0.389	0.943
Neurological Surgery	0.0	0.530	0.601	0.494	0.965
Orthopaedic Surgery	0.0	0.576	0.576	0.650	0.979

The learned policy exhibits significant performance disparities across clinical and demographic subgroups, indicating uneven generalization from the training data. Performance varies substantially by medical specialty (Table 1). The agent shows moderate success in common scenarios such as plastic surgery (28.6% solve rate) and vascular surgery (16.0% solve rate), but struggles or fails altogether in less-represented specialties such as neurological surgery and orthopaedic surgery, both of which had a 0% solve rate across all runs. These gaps suggest the policy has overfit to case patterns prevalent in the training data distribution.

Table 2: Performance breakdown by patient demographics, averaged over 3 seeds. Asterisks (*) denote a statistically significant difference ($p < 0.05$) across groups for that metric, determined by a one-way ANOVA test.

Group	Solved (%)	F1 Score	Recall	Precision	Specificity
Gender					
Male	13.2	0.534	0.667	0.497	0.963
Female	11.8	0.454*	0.622	0.401*	0.957
Age Group					
0-18	9.1	0.544	0.666	0.538	0.974
19-40	8.6	0.503	0.652	0.461	0.963
41-65	21.1	0.534	0.659	0.485	0.975
65+	6.7	0.430	0.634	0.359	0.911*
Race					
Black	15.1	0.521	0.668	0.470	0.961
Caucasian	15.0	0.531	0.673	0.501	0.976
Asian	6.9	0.465	0.607	0.430	0.951

Critically, the agent displays statistically significant biases across demographic groups (Table 2). A significant performance gap exists between genders, with male patient cases showing higher diagnostic F1 score and precision ($p < 0.05$). The most severe bias relates to age: The agent

performs best on middle-aged patients (41-65), achieving its highest solve rate (21.1%). In contrast, performance on geriatric patients (65+) collapses, with a significantly lower total reward ($p < 0.05$) and a dramatic, highly significant drop in specificity to 0.911 ($p < 0.001$). This indicates that the policy is not only ineffective but also potentially unsafe for this demographic, as it is more likely to select harmful actions. Performance differences across racial groups were not statistically significant.

4.3 Ablation Studies

We conducted ablation studies to isolate the contributions of the reward function, state representation model, and training data size. Further ablations on hyperparameter sensitivity and action space composition are presented in Appendix A.

Table 3: Ablation of reward function design.

	Overall Performance			Diagnostic Accuracy			
Reward Model	**Reward**	**Solved (%)**	**Steps**	**Recall**	**Precision**	**F1 Score**	**Specificity**
Our Method	1.013	**20.6**	17.79	0.673	0.449	0.499	0.959
Entrust Scaling	6.191	17.6	**17.71**	**0.684**	**0.460**	**0.511**	0.967
Zero-Clipped Scaling	**6.544**	17.6	17.97	0.588	0.394	0.427	0.931
Score Agnostic	4.706	17.6	18.09	0.676	0.432	0.483	**0.970**

Reward Function. The reward structure critically shapes the policy's trade-offs between diagnostic accuracy, completion rate, and safety (Table 3). Directly using normalized EPA scores ("Entrust Scaling") yielded the highest F1 Score (0.511), while our method's terminal rewards produced a +3% higher case completion rate. Nullifying penalties ("Zero-Clipped Scaling") degraded all accuracy metrics, confirming their importance in policy shaping. Conversely, a score-agnostic model induced the most conservative policy, achieving the highest specificity (0.970).

Table 4: Ablation of foundation model for state embeddings.

	Overall Performance			Diagnostic Accuracy			
Foundation Model	**Reward**	**Solved (%)**	**Steps**	**Recall**	**Precision**	**F1 Score**	**Specificity**
`Bio_ClinicalBERT` [2]	1.013	**20.6**	17.79	0.673	0.449	0.499	0.959
`ClinicalBERT` [17, 33]	**1.862**	**20.6**	**17.59**	0.697	**0.501**	**0.542**	0.962
`BioBERT v1.1` [15]	1.038	17.6	**17.59**	0.675	0.474	0.519	0.960
`BERT-base` [6]	0.899	17.6	17.91	0.671	0.446	0.497	0.961
`Qwen3-Embedding-0.6B` [36]	1.227	**20.6**	17.76	0.682	0.456	0.506	0.958
`Qwen3-Embedding-4B` [36]	1.732	17.6	17.85	**0.706**	0.467	0.522	**0.964**

Foundation Model. Using domain-specific language models for state representation significantly improves performance (Table 4). `ClinicalBERT`, pretrained on clinical notes, achieved the highest F1 score (0.542), a +4.5 percentage-point improvement over the general-purpose `BERT-base`. The larger `Qwen3-Embedding-4B` model also performed strongly, attaining the highest recall (0.706). While our framework is robust to the choice of encoder, these results confirm that performance is enhanced by semantic representations aligned with the clinical domain.

Table 5: Ablation of training data fraction.

	Overall Performance			Diagnostic Accuracy			
Data Fraction	**Reward**	**Solved (%)**	**Steps**	**Recall**	**Precision**	**F1 Score**	**Specificity**
10%	-9.293	8.8	18.71	0.485	0.315	0.350	0.929
25%	-1.610	**20.6**	17.62	0.635	0.421	0.467	0.948
50%	-1.084	14.7	17.79	0.666	0.459	0.504	0.950
75%	0.580	**20.6**	**17.38**	0.675	**0.455**	**0.506**	0.954
100% (Full)	**1.013**	**20.6**	17.79	**0.673**	0.449	0.499	**0.959**

Data Regime. Policy performance scales with the quantity of training data, but with diminishing returns (Table 5). The most substantial gain occurs when increasing data from 10% to 50%, boosting the F1 score by over 15 percentage-points (from 0.350 to 0.504). Performance plateaus beyond this point, suggesting that either model capacity is reached or the additional data provides insufficient novelty to improve generalization further. The low reward on smaller data fractions is driven by the agent's failure to solve cases, thus incurring large terminal penalties.

4.4 Qualitative Analysis of Agent Behavior

Analysis of individual trajectories reveals the agent learns effective templates for common scenarios but lacks deeper contextual reasoning.

Goal-Oriented Policy with Contextual Oversights. In a standard preoperative workup, the agent achieved a recall of 1.0 but a precision of only 0.56. It successfully executed a learned template for the primary task but ordered a battery of irrelevant tests based on a secondary feature (family history of breast cancer). This behavior suggests that the policy relies on high-level pattern matching but fails to weigh the relevance of different state features, resulting in a comprehensive but inefficient "checklist-style" execution.

Strategy Drift from Lack of Long-Term Coherence. In a more complex case of an adrenal incidentaloma, an initially coherent diagnostic strategy devolved into irrelevant actions, including ordering a breast cancer workup for a male patient. This catastrophic failure indicates a brittle policy that relies on superficial keyword correlations (e.g., "nodule") without integrating critical context like patient gender. This highlights a failure to develop a robust, causal model of the diagnostic process, leading to a breakdown in long-term decision-making.

4.5 Downstream Evaluation of Clinical Reasoning

To assess the clinical utility of the information gathered by the agent, we evaluated its trajectory on a downstream question-answering task. The agent's policy provided negligible informational gain for this task. An external LLM achieved an accuracy of 66.34% using the agent's action summary as context, a marginal improvement of less than 0.4% compared to baselines with no actions (66.02%) or random actions (65.37%). This suggests the agent's policy does not gather information that significantly enhances performance on this specific reasoning task. Furthermore, providing the oracle set of all positive actions yielded the lowest accuracy (64.72%), suggesting that the QA task primarily tests reasoning based on the initial patient presentation, and that additional diagnostic results, even optimal ones, may act as distracting context for the LLM in this specific evaluation format.

5 Discussion

The findings show that an RL agent with a natural-language state can acquire a stable and competent policy for EPA-style simulations, as reflected in steadily rising returns and strong recall with high specificity (Figure 1). At the same time, its comparatively lower precision indicates a preference for breadth over parsimony: the policy tends to accumulate many diagnostically neutral actions while reliably capturing required ones. The qualitative trajectories reinforce this picture. In common scenarios the agent executes a reliable template, yet it is less adept at pruning actions based on evolving case context, which manifests as a conservative, checklist-oriented strategy. This behavior is consistent with the incentives in our environment: a small step penalty, immediate utility rewards, and a sizable terminal bonus collectively favor comprehensive coverage with limited pressure to optimize marginal utility once likely positives have been identified.

Performance heterogeneity across specialties and demographics (Tables 1 and 2) suggests uneven generalization rather than a universal deficit. Where the training distribution is richer, the policy performs reasonably; where cases are sparser or atypical, solve rates decline and error profiles shift. The drop in specificity for geriatric patients is particularly important, as it indicates a higher propensity to select low-value or potentially harmful actions in that subgroup. These patterns point to distributional imbalance as a primary driver and highlight the need for fairness-aware training objectives, subgroup-sensitive validation, and targeted data augmentation to reduce gap amplification.

We view this not as a fundamental limitation of RL for clinical simulation, but as a reminder that agent objectives and data curricula must be designed with equity and safety in mind.

The downstream QA analysis provides a complementary lens. Minimal gains from using the agent's trajectories as context indicate that "solving the simulation" is not synonymous with organizing information in a way that benefits separate reasoning tasks. The agent appears to collect the right pieces frequently (high recall) but does not consistently assemble them into a compact, decision-supportive narrative for an external model. Bridging this gap likely requires objectives that explicitly value informativeness and coherence (e.g., penalizing redundant evidence, rewarding discriminative findings, or training a summarization head that learns to produce compact clinical state descriptions aligned with downstream tasks).

Several design choices constrain what the agent can learn. The simulated environment abstracts a complex clinical workflow into a discrete action set with oracle-derived utilities, which necessarily simplifies real-world trade-offs among benefit, risk, cost, and time. The language representation aggregates sequential text into fixed vectors, which is efficient but may blur temporal dependencies and attenuate rare but crucial signals. Finally, the reward emphasizes task completion and aggregate utility more than calibrated decision quality, leaving limited capacity for the agent to express uncertainty or defer.

These observations motivate directions for future work. Architecturally, adding memory and attention (e.g., Transformer-based critics or recurrent policies) can preserve temporal structure and support long-range credit assignment. Framing the problem as a partially observed process with explicit belief states would allow the agent to quantify and act on uncertainty, improving both efficiency and safety. On the objective side, cost- and risk-sensitive rewards, counterfactual or inverse RL from expert trajectories, and constraints that penalize over-testing can better align incentives with clinical priorities. To address subgroup disparities, distributionally robust optimization, reweighting, and stratified curricula—paired with pre-specified fairness metrics and guardrails—can help stabilize performance across underrepresented cases. Finally, evaluation should extend beyond solve rates and F1 to include calibration, justification quality, and human-in-the-loop assessments, as well as tasks that test whether the agent's information-gathering improves other clinically relevant computations (e.g., differential diagnosis ranking or indication-specific decision rules). Taken together, these refinements aim to shift the agent from procedural mimicry toward context-sensitive, uncertainty-aware reasoning while keeping safety and equity central.

6 Conclusion

This study presents a natural language–based RL framework for surgical EPA simulations and demonstrates that a masked-PPO agent can learn a functional, stable policy with strong recall and high specificity across diverse patient cases. The same experiments surface key limitations that are instructive for future work: reduced precision stemming from conservative, checklist-style behavior; uneven generalization across specialties and demographics; and limited transfer of the collected information to a separate reasoning task. These results distinguish success at the simulation objective from the broader goal of cultivating clinically useful reasoning.

Methodologically, the work contributes a clear formulation of clinical EPA simulations as an MDP with semantic state representations, a practical masking mechanism for constrained actions, and an evaluation suite that combines aggregate metrics, subgroup analysis, and qualitative trajectory inspection. Looking ahead, we see the most leverage in (i) architectures that retain temporal structure and model uncertainty; (ii) objectives that explicitly trade off benefit, risk, cost, and informativeness, potentially learned from expert behavior; (iii) fairness-aware training and validation to stabilize performance across subgroups; and (iv) evaluation protocols that test whether agent-driven information-gathering improves downstream clinical tasks and supports well-calibrated, interpretable decisions. With these extensions, language-grounded RL has the potential to evolve from a capable simulator policy into a foundation for educational tools and decision-support systems in healthcare settings.

References

[1] Alaa Awad Abdellatif, Naram Mhaisen, Amr Mohamed, Aiman Erbad, and Mohsen Guizani. Reinforcement learning for intelligent healthcare systems: A review of challenges, applications, and open research issues. *IEEE Internet of Things Journal*, 10(24):21982–22007, 2023. doi: 10.1109/JIOT.2023.3288050.

[2] Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, et al. Publicly available clinical BERT embeddings, 2019.

[3] Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, et al. Healthbench: Evaluating large language models towards improved human health, 2025.

[4] Suhana Bedi, Hejie Cui, Miguel Fuentes, Alyssa Unell, Michael Wornow, Juan M. Banda, et al. Medhelm: Holistic evaluation of large language models for medical tasks, 2025.

[5] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, et al. Openai gym, 2016.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding, 2018.

[7] Charles R. Harris, K. Jarrod Millman, Stéfan van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, et al. Array programming with numpy, 2020.

[8] Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, et al. Deep reinforcement learning with a natural language action space. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1621–1630, 2016. doi: 10.18653/v1/P16-1153.

[9] Wei Hu, Lechao Xiao, and Jeffrey Pennington. Provable benefit of orthogonal initialization in optimizing deep linear networks, 2020.

[10] Marie Humbert-Droz, Pritam Mukherjee, and Olivier Gevaert. Strategies to address the lack of labeled data for supervised machine learning training with electronic health records: Case study for the extraction of symptoms from clinical notes. *JMIR Med Inform*, 10(3):e32903, 2022. doi: 10.2196/32903.

[11] Pushkala Jayaraman, Jacob Desman, Moein Sabounchi, Girish N Nadkarni, and Ankit Sakhuja. A primer on reinforcement learning in medicine for clinicians. *npj Digital Medicine*, 7(1):337, 2024. doi: 10.1038/s41746-024-01316-0.

[12] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021. doi: 10.3390/app11146421.

[13] Kia Khezeli, Scott Siegel, Benjamin Shickel, Tezcan Ozrazgat-Baslanti, Azra Bihorac, and Parisa Rashidi. Reinforcement learning for clinical applications. *Clinical Journal of the American Society of Nephrology*, 18(5):521–523, 2023. doi: 10.2215/CJN.0000000000000084.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[15] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, et al. Biobert: a pre-trained biomedical language representation model for biomedical text mining, 2019.

[16] Dana T Lin, Edward F Melcer, Oleksandra Keehl, Hyrum Eddington, Amber W Trickey, Jason Tsai, et al. Entrust: A serious game-based virtual patient platform to assess entrustable professional activities in graduate medical education. *Journal of Graduate Medical Education*, 15(2):228–236, 2023. doi: 10.4300/JGME-D-22-00518.1.

[17] Xiaohong Liu, Hao Liu, Guoxing Yang, Zeyu Jiang, Shuguang Cui, Zhaoze Zhang, et al. A generalist medical language model for disease diagnosis assistance. *Nature Medicine*, 2025. doi: 10.1038/s41591-024-03416-6.

[18] Brielen Madureira and David Schlangen. An overview of natural language state representation for reinforcement learning, 2020.

[19] Jeff Mitchell and Mirella Lapata. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429, 2010. doi: 10.1111/j.1551-6709.2010.01106.x.

[20] Volodymyr Mnih, Adri‘a Puigdom‘enech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, et al. Asynchronous methods for deep reinforcement learning, 2016.

[21] Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. Language understanding for text-based games using deep reinforcement learning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1–11, 2015. doi: 10.18653/v1/D15-1001.

[22] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174, pages 248–260, 2022. doi: 10.48550/arXiv.2203.14371.

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, et al. Pytorch: An imperative style, high-performance deep learning library, 2019.

[24] Fabian Pedregosa, Ga"el Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, et al. Scikit-learn: Machine learning in python, 2012.

[25] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[26] Samuel Schmidgall, Rojin Ziaei, Carl Harris, Eduardo Reis, Jeffrey Jopling, and Michael Moor. Agentclinic: a multimodal agent benchmark to evaluate ai in simulated clinical environments, 2025.

[27] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[28] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018.

[29] Erez Schwartz, Guy Tennenholtz, Chen Tessler, and Shie Mannor. Language is power: Representing states using natural language in reinforcement learning, 2019.

[30] Shefaly Shorey, Tang Ching Lau, Siew Tiang Lau, and Emily Ang. Entrustable professional activities in health care education: a scoping review. *Medical Education*, 53(8):766–777, 2019. doi: 10.1111/medu.13879.

[31] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, et al. Gemma 3 technical report, 2025.

[32] Olle ten Cate. Nuts and bolts of entrustable professional activities. *Journal of Graduate Medical Education*, 5(1):157–158, 2013. doi: 10.4300/JGME-D-12-00380.1.

[33] Guangyu Wang, Xiaohong Liu, Zhen Ying, Guoxing Yang, Zhiwei Chen, Zhiwen Liu, et al. Optimized glycemic control of type 2 diabetes with reinforcement learning: a proof-of-concept trial. *Nature Medicine*, 29(10):2633–2642, 2023. doi: 10.1038/s41591-023-02552-9.

[34] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 2020. doi: 10.18653/v1/2020.emnlp-demos.6.

[35] Chao Yu, Jiming Liu, and Shamim Nemati. Reinforcement learning in healthcare: a survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2020. doi: 10.1145/3477600.

[36] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025.

Appendices

A Additional Ablations

We conduct further ablations to analyze the impact of key hyperparameters and the composition of the action space on agent performance.

A.1 Hyperparameter Sensitivity

We assess the agent's sensitivity to three key hyperparameters: learning rate, entropy coefficient, and rollout buffer size (n_{steps}). The baseline configuration (lr=3e-4, ent=0.05, n_steps=10240) provides a strong balance of performance. As shown in Table 6, performance degrades with very high or very low learning rates. A smaller entropy coefficient of 0.01 yields the strongest performance, indicating the policy benefits from a reduced, but non-zero, incentive for exploration compared to the baseline. The model is largely robust to changes in the rollout buffer size.

Table 6: Hyperparameter ablation results on the evaluation set.

		Overall Performance			Diagnostic Accuracy			
Parameter	**Value**	**Reward**	**Solved (%)**	**Steps**	**Recall**	**Precision**	**F1**	**Specificity**
Learning Rate	1e-5	-6.971	8.8	18.71	0.511	0.316	0.358	0.947
	1e-4	-2.221	11.8	18.29	0.597	0.394	0.436	0.957
	3e-4	**1.013**	**20.6**	17.79	**0.673**	0.449	**0.499**	**0.959**
	1e-3	-0.466	17.6	**17.44**	0.638	**0.452**	0.490	0.954
	1e-2	-9.965	11.8	18.50	0.410	0.219	0.260	0.922
Entropy Coef.	0	1.114	17.6	17.88	**0.694**	0.459	0.509	0.962
	0.01	**1.956**	**23.5**	**17.50**	0.685	**0.472**	**0.517**	**0.963**
	0.05	1.013	20.6	17.79	0.673	0.449	0.499	0.959
	0.1	0.087	14.7	17.85	0.661	0.457	0.501	0.959
	0.2	0.730	17.6	17.85	0.675	0.460	0.507	0.960
Rollout Buffer	2560	0.514	14.7	18.00	0.678	0.458	0.504	0.956
	5120	0.634	17.6	17.71	0.662	0.465	0.506	0.959
	10240	1.013	**20.6**	17.79	0.673	0.449	0.499	0.959
	15360	**1.487**	17.6	**17.56**	**0.712**	0.470	**0.525**	**0.960**
	20480	0.498	17.6	17.65	0.672	**0.473**	0.515	**0.960**

A.2 Action Space Composition

We investigate the agent's reliance on different categories of clinical actions through three experiments: restricting the agent to a single category (Table 7), excluding one category at a time (Table 8), and cumulatively adding categories (Table 9).

Restricted Action Space ('Only'). When limited to a single action category, the agent's performance reveals the intrinsic utility of each type. Categories with simple, universally positive actions (e.g., Oxygen, Fluids, Consult) lead to 100% solve rates on applicable cases, though their low precision reflects a narrow scope. In contrast, information-gathering categories like Lab Tests and Interventions yield higher F1 scores, demonstrating their broader diagnostic value.

Table 7: Performance when the agent is restricted to a single action category.

	Overall Performance			Diagnostic Accuracy			
Category Only	**Reward**	**Solved (%)**	**Steps**	**Recall**	**Precision**	**F1**	**Specificity**
Lab Tests	8.781	67.6	10.41	0.928	**0.592**	**0.666**	0.978
Imaging	5.918	73.5	7.82	0.869	0.399	0.451	0.966
Interventions	11.095	94.1	5.62	0.983	0.483	0.550	0.980
Medications	8.737	88.2	7.59	0.928	0.254	0.311	0.968
Blood Supplement	14.249	**100.0**	1.15	**1.000**	0.162	0.176	0.995
Consult	14.707	**100.0**	1.50	**1.000**	0.281	0.307	0.994
Fluids	14.599	**100.0**	**1.09**	**1.000**	0.088	0.088	0.996
Oxygen	**14.834**	**100.0**	**1.09**	**1.000**	0.118	0.127	**1.000**

Leave-One-Out Exclusion ('Exclude'). Removing a single action category tests policy robustness. Excluding 'Medications' improves performance, suggesting the agent struggles to use these actions effectively and their absence simplifies the task. Conversely, excluding 'Lab Tests' significantly harms the F1 score, confirming their critical role in the diagnostic process.

Table 8: Performance when a single action category is excluded from the action space.

	Overall Performance			Diagnostic Accuracy			
Category Excluded	**Reward**	**Solved (%)**	**Steps**	**Recall**	**Precision**	**F1**	**Specificity**
Baseline (None)	1.013	20.6	17.79	0.673	0.449	0.499	0.959
Lab Tests	-1.453	20.6	17.06	0.715	0.301	0.390	0.945
Imaging	2.353	23.5	16.97	0.762	0.467	0.541	0.958
Interventions	0.314	20.6	17.24	0.732	0.415	0.484	0.956
Medications	**3.074**	**32.4**	**16.32**	**0.783**	**0.489**	**0.558**	0.959
Blood Supplement	0.677	14.7	17.82	0.673	0.470	0.512	0.962
Consult	-0.253	14.7	17.91	0.684	0.460	0.509	0.956
Fluids	1.468	20.6	17.24	0.699	0.469	0.524	**0.964**
Oxygen	1.044	17.6	17.62	0.687	0.480	0.524	0.960

Cumulative Addition ('Add'). As action categories are cumulatively added, performance initially drops. Starting with only high-utility 'Interventions' is easy, but as more complex, lower-utility actions ('Blood Supplement', 'Consult') are introduced, the agent's task becomes harder, leading to lower rewards and solve rates. Performance stabilizes as the full action space is restored, indicating the agent learns to manage the complexity.

Table 9: Performance as action categories are cumulatively added.

	Overall Performance			Diagnostic Accuracy			
Cumulative Actions Added	**Reward**	**Solved (%)**	**Steps**	**Recall**	**Precision**	**F1**	**Specificity**
Interventions	**11.095**	**94.1**	**5.62**	**0.983**	0.483	0.550	**0.980**
+ Lab Tests	3.623	41.2	14.35	0.858	**0.507**	**0.593**	0.963
+ Imaging	3.280	38.2	15.68	0.810	0.486	0.564	0.961
+ Medications	0.913	20.6	17.35	0.700	0.465	0.517	0.964
+ Blood Supplement	-0.068	17.6	17.79	0.695	0.456	0.507	0.958
+ Consult	0.561	20.6	17.32	0.675	0.450	0.499	0.963
+ Fluids	1.044	17.6	17.62	0.687	0.480	0.524	0.960
+ Oxygen (Full)	1.013	20.6	17.79	0.673	0.449	0.499	0.959

B Downstream QA Details

To evaluate the clinical utility of the information gathered by our agent, we designed a downstream question-answering (QA) task. For each case in the test set, we prompted an external Large Language Model (LLM) to answer multiple-choice questions based on the clinical scenario.

Model and Task Setup We used `Gemma 3 27B-IT` as the external reasoning agent. The evaluation was conducted under four distinct conditions to isolate the informational value of the agent's actions:

1. **RL Agent Trajectory:** The LLM was provided with the patient context (if required by the question) and the sequence of actions selected by our trained RL agent.
2. **Random Actions Baseline:** The LLM was provided with the patient context and a sequence of randomly selected, valid actions. The number of random actions was identical to the number of actions taken by our RL agent for that specific case.
3. **No Actions Baseline:** The LLM was provided with only the patient context, without any information about actions taken. This measures the LLM's ability to answer based solely on the initial case presentation.
4. **All Positive Actions (Oracle):** The LLM was provided with the patient context and the complete set of all clinically appropriate (non-negative utility) actions for the case. This condition serves as an oracle to test the effect of providing maximal, correct information.

Prompt Format A consistent prompt structure was used for all three conditions. The prompt specified an expert persona, provided the relevant context and actions (if any), stated the question and options, and instructed the model to return only the full text of the correct answer. The specific format is shown below.

```
You are an expert medical professional. Based on the
provided information, answer the multiple-choice question.
---
CONTEXT:
{Patient Information String}
---
STEPS TAKEN / ACTIONS ORDERED:
{Action 1, Action 2, ...}
---
QUESTION:
{Question Text}

OPTIONS:
- {Answer Option A}
- {Answer Option B}
- {Answer Option C}
---
INSTRUCTION: Choose the best answer from the options above.
Respond with ONLY the full text of the correct answer, without
any prefixes or explanations.
```

Note that the 'CONTEXT' and 'STEPS TAKEN' blocks were conditionally included based on the question's requirements and the specific evaluation condition being tested.

Evaluation An answer was marked as correct if the LLM's generated text contained a case-insensitive, punctuation-normalized match for the ground-truth answer string.

C Dataset and Environment Details

Cases. Each case is a JSON object with: `caseId`, free-text `patientInformation`, numeric `initialVitals` (dbp, hr, rr, sbp, spo2, temp), free-text per-system `initialPhysicalExam`, and a list `caseOrders` where each item has `fullName` (action), `result` (free text), and utility scores (`score`, `entrustScore`, `zeroClippedScore`). Optional multiple-choice `questions` are used only for a downstream QA probe (not for RL).

Specialty labels. Specialties are assigned by prompting a Gemma model to map each case (full context) to one of the fourteen American College of Surgeons surgical specialties.

Split and action coverage. An 80/20 random split creates train/test. To avoid unseen actions at test time, each test case is filtered to retain only actions that appear in the training split.

Numeric features & parsing. Vitals are always present as keys {hr, rr, spo2, sbp, dbp, temp}. Additional numeric values are parsed from order `result` text using three pattern types: (i) keyed ranges ("Sodium: 135–145"), (ii) keys-only lists, and (iii) value-only strings (mapped to the action name as a key when appropriate). For each key, mean/std are computed over the training split; observed values are z-scored online.

Text embeddings. Two sources are embedded: (i) initial case text (patient summary + all initial exam strings concatenated) and (ii) per-(case, action) `result` texts. We use Hugging Face `AutoTokenizer`/`AutoModel` with Bio_ClinicalBERT as default; token-level last hidden states are mean-pooled with attention masking, then L2-normalized (Transformers [34], Bio_ClinicalBERT [2]). Embeddings are cached in a single NPZ per encoder, keyed by SHA1(text), and reused across runs.

MDP & observation. Finite-horizon MDP with $T_{\max} = 20$. The observation at step t is $[\, e_{\text{init}} \,\|\, e_{\text{hist},t} \,\|\, v_{\text{labs},t} \,\|\, \tau_t \,]$: e_{init} is the fixed initial-text embedding; $e_{\text{hist},t}$ is the L2-normalized running average of embeddings of all revealed `result` texts; $v_{\text{labs},t}$ is the z-scored numeric vector over the learned lab/vital key set; $\tau_t = t/T_{\max}$. Dimension $= 2d_{\text{emb}} + d_{\text{lab}} + 1$.

Actions, feasibility, termination. The global action set is the sorted unique `caseOrders[*].fullName`. A dynamic mask enables only case-valid, not-yet-selected actions at each step. Episodes terminate when all positive-utility actions for the case have been taken or when $t = T_{\max}$.

Rewards. Default "smart" reward: per-step -0.2; immediate $+$`entrustScore`$/100$ if available; terminal bonus on solve $+10{+}5\,(1{-}t/T_{\max})$; terminal penalty on timeout $-10\,(1{-}\text{Recall})$. Alternatives used in ablations: (i) *Entrust* (immediate `entrustScore`$/100$ only), (ii) *Zero-Clipped* (immediate $\max(0,$ `entrustScore`$/100)$ only), (iii) *Score-agnostic* ($+1/-1/0$ for positive/negative/neutral utilities).

Evaluation metrics. On termination we compute: solved indicator, steps, total reward, recall/precision/F1 against positive-utility actions, specificity (fraction of negative-utility actions avoided), counts of positive/negative/neutral actions taken, completion-speed $(T_{\max} - t)/T_{\max}$, and the ordered action sequence.

D Implementation Details

Core stack. PyTorch for tensors [23], NumPy for arrays [7], Transformers for encoders [34], scikit-learn for the split [24], Gymnasium for the environment API [5], Stable-Baselines3 and SB3-Contrib for PPO and action masking (`MaskablePPO`, `ActionMasker`) [25]. PPO/GAE follow prior work [27, 28]; shared-MLP actor-critic follows prior work [20]; Adam optimizer [14].

Policy/algorithm. Algorithm: Masked PPO with dynamic action masks applied throughout training and evaluation. **Network:** shared actor–critic MLP (two hidden layers of 64, `tanh`, orthogonal init). **Key hyperparameters (defaults):** learning rate 3×10^{-4}; entropy coefficient 0.05; PPO epochs 2; minibatch size 64; discount $\gamma = 0.99$; GAE-$\lambda = 0.95$; clip $\epsilon = 0.2$.

Seeding & device. A single integer seed is set for Python `random`, NumPy, PyTorch, and SB3; device is CUDA if available, else CPU. On average, one training and evaluation experiment took 25 minutes. All ablations and the downstream QA evaluation together took approximately 9 hours.

E Broader Societal Impact

The primary societal impact of this research is methodological. It provides a framework for rigorously studying and identifying failure modes in automated clinical reasoning agents well before any

real-world deployment. The key contribution in this regard is the clear demonstration of performance disparities across demographic subgroups, particularly the reduced safety profile in geriatric cases. This finding provides concrete evidence that standard RL objectives, when applied to imbalanced clinical data, can produce policies that amplify societal biases. By surfacing these fairness and generalization challenges within a controlled simulation, this work underscores the necessity of developing fairness-aware learning objectives and robust evaluation protocols as foundational prerequisites for any future translation of such technologies.

F Licenses

All assets are credited to their original creators. The licenses for third-party assets used in this work are listed below. The primary clinical case dataset used for training and evaluation is proprietary and not publicly available.

- **PyTorch** [23]: Deep learning framework used for model implementation. License: BSD-style.
- **NumPy** [7]: Library for numerical operations. License: BSD 3-Clause.
- **Transformers** [34]: Library for accessing pre-trained models and tokenizers. License: Apache 2.0.
- **scikit-learn** [24]: Used for data splitting. License: BSD 3-Clause.
- **Gymnasium** [5]: API for the reinforcement learning environment. License: MIT.
- **Stable-Baselines3** [25]: Library for PPO implementation and training. License: MIT.
- **Bio_ClinicalBERT** [2]: Pre-trained language model used for default state embeddings. License: Apache 2.0.
- **ClinicalBERT** [17, 33]: Used in ablation studies for state embeddings. License: Apache 2.0.
- **BioBERT v1.1** [15]: Used in ablation studies for state embeddings. License: Apache 2.0.
- **BERT-base** [6]: Used in ablation studies for state embeddings. License: Apache 2.0.
- **Qwen3 Embeddings** [36]: Used in ablation studies for state embeddings. License: Apache 2.0.
- **Gemma Models** [31]: Used for the downstream QA task and specialty labeling. License: Gemma Terms of Use.

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 stated contributions match the methods and evidence presented in Sec.3 and Sec.4 (Fig.1, Tables1–5).

   Guidelines:

   - The answer NA 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 NA 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: Limitations and failure modes are discussed in Sec.5 and evidenced by subgroup gaps in Sec.4 (Tables1, 2).

   Guidelines:

   - The answer NA 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: [NA]

Justification: Not applicable—no new theoretical results are claimed; the work is empirical (see Sec.3 and Sec.4).

Guidelines:

- The answer NA 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: Key details are specified in Sec.3, Alg.1, Appendices C, and D.

   Guidelines:

   - The answer NA 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: [No]

Justification: The primary dataset is proprietary and code is not released; see Appendices C and F for data description and terms.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Training/test setup, $T_{\max}$, masks, optimizer, hyperparameters, and seeds are detailed in Sec. 3 and Appendix D.

Guidelines:

- The answer NA 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]

Statistical significance is reported via one-way ANOVA with p-values for demographics (Table 2); variability across settings appears in ablation tables (e.g., Tables 6–9).

Guidelines:

- The answer NA 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: Compute device and runtimes are provided in Appendix D.

   Guidelines:

   - The answer NA 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: We have read the NeurIPS Code of Ethics and made sure that the paper conforms to it.

   Guidelines:

   - The answer NA 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: Positive/negative impacts (e.g., simulation safety benefits vs. subgroup disparities) are discussed in Sec. 5 and Appendix E.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No public release of high-risk models or scraped datasets.

    Guidelines:

    - The answer NA 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: Third-party assets and licenses are enumerated in Appendix F and cited in the references.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: No new public assets are released.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: No crowdsourcing or human-subjects studies; experiments use simulated cases.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: No human-subjects research or identifiable data.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLM usage is described for downstream QA and specialty labeling in Appendices B and C.

    Guidelines:

    - The answer NA 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 (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.