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 $+\texttt{entrustScore}/100$ 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 $\texttt{entrustScore}/100$ only), (ii) *Zero-Clipped* (immediate $\max(0, \texttt{entrustScore}/100)$ 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.