MEDMDT: SELF-EVOLVING MULTIDISCIPLINARY AGENT TEAMS FOR CLINICAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Clinical reasoning over electronic health records (EHRs) requires integrating heterogeneous longitudinal evidence, resolving competing hypotheses, and learning from experience without observing future events. We present MedMDT (medical multidisciplinary agent teams), an event-driven system that couples task-routed specialists with a coordinating chief and an evidence critic. Its central design is a dual-state adaptation protocol: Decentralized Low-Rank Policy Optimization (D-LRPO) learns role-specific parametric policies from reward-scored rollouts, whereas Critic-Bandit Skill Policy Optimization (CB-SPO) selects, validates, and versions reusable textual policies. To eliminate data leakage, the architecture isolates adaptation, inference, and evaluation through strict temporal filtering, hidden labels, and frozen benchmarks. We evaluate MedMDT on MedEHR, comprising 5,100 task instances spanning common, rare, cross-dataset, and randomized clinical trial (RCT) settings. With a shared chief backbone, MedMDT attains 0.4730 Sample-F1 on MedEHR-Common, compared with 0.4108 for the strongest same-chief baseline. MedMDT Pro achieves the highest MedEHR-RCT average among the compared systems (0.6530) while remaining competitive on the other subsets. Increasing adaptation data from 600 to 3,000 cases raises Sample-F1 from 0.4730 to 0.5699 on MedEHR-Common and from 0.6126 to 0.6532 on MedEHR-RCT. Component ablations and prediction-transition analyses support complementary contributions from specialization, critique, and the two adaptation channels. Code will be released on GitHub.

1 INTRODUCTION

Electronic health records (EHRs) distribute clinically relevant information across diagnoses, laboratory measurements, microbiology, medications, procedures, and patient movements. Reasoning over these records is a sequential evidence problem: an agent must reconstruct what was available at a decision time, retrieve the observations relevant to a task, and map uncertain conclusions to a clinically meaningful vocabulary. Irregular sampling, incomplete observations, long histories, and changing care patterns make this substantially different from answering a self-contained medical question (Li et al., 2023; 2026b; Shan et al., 2025; Li et al., 2025). EHR-agent benchmarks increasingly expose these demands through direct interaction with patients data (Liao et al., 2026).

A useful design analogy is the multidisciplinary clinical team. Specialists examine evidence through distinct task perspectives, a coordinating clinician reconciles their proposals, and explicit review can expose unsupported conclusions or missing evidence. Existing medical committees and multi-agent debate demonstrate the value of collaboration (Tang et al., 2024; Kim et al., 2024; Du et al., 2023), but discussion alone neither assigns persistent expertise nor determines how experience should update different responsibilities. Conversely, adapting one monolithic policy entangles specialization, coordination, and error correction. As illustrated in Figure 1, we ask: *can a clinical agent team improve both role-specific behavior and reusable reasoning rules while preserving a strict boundary among historical evidence, post-decision feedback, and evaluation?*

We introduce MedMDT, a process-based agent architecture with typed messages linking an EHR retriever, a chief, specialist agents, and a critic. The chief first forms a private patient summary and then integrates task-routed proposals; the critic identifies supported, unsupported, and missing candidates before the final decision is committed. Five standard EHR tasks invoke one domain spe-

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

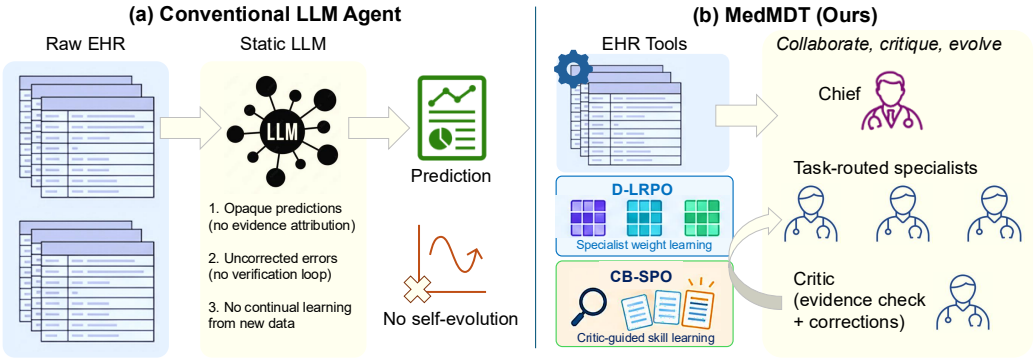


Figure 1: Comparison between a conventional LLM agent and proposed MedMDT. MedMDT augments the workflow with task-routed specialists, evidence critique, and two policies: D-LRPO (Decentralized Low-Rank Policy Optimization) and CB-SPO (Critic-Bandit Skill Policy Optimization).

cialist followed by critique and chief normalization, whereas Procedures and all RCT tasks invoke multiple specialists and require cross-proposal reconciliation. Across cases, MedMDT evolves a factorized team state: D-LRPO assigns separate low-rank parameters to specialist roles, whereas CB-SPO maintains compact, versioned textual skills. The novelty is an executable ownership and synchronization protocol: parametric and semantic state have different owners, acceptance tests, and rollback paths, yet receive a shared reward only after an immutable team prediction has been committed. This turns a transient discussion into an auditable learning system while preserving the boundary between inference, feedback, and evaluation.

Our contributions are threefold. First, we formulate an event-driven multidisciplinary reasoning protocol with explicit evidence horizons, typed communication, label-private inference, and commit-before-feedback semantics. Second, we introduce synchronized dual-state evolution: role-local low-rank policies capture specialist behavior, while a bounded textual policy captures reusable and auditable reasoning rules; validation freezes, channel-specific ownership, and versioned rollback limit interference. To our knowledge, prior clinical-agent work has not combined these two persistent, separately governed adaptation states within one commit-before-feedback team protocol. Third, we construct four complementary MedEHR evaluation settings and provide task-level comparisons, adaptation trajectories, component ablations, disagreement analysis, and prediction-transition analysis. Across three independent runs, the full configuration obtains 0.4730 ± 0.0045, improving over the frozen full team by 0.0438; the D-LRPO × CB-SPO interaction remains positive after multiple-comparison correction ($+0.0089$, $p_{\text{adj}} = 0.0228$). Together, these experiments test endpoint accuracy, the source of corrections, and behavior as adaptation data accumulate, while separating this evidence from causal or deployment claims.

2 Related Work

2.1 Agentic Systems

Tool-using language agents interleave reasoning and actions, as exemplified by ReAct (Yao et al., 2022) and ReSum (Wu et al., 2025). Multi-agent debate improves some reasoning tasks by iteratively comparing independent responses (Du et al., 2023). In medicine, MedAgents organizes multidisciplinary discussion without parameter updates (Tang et al., 2024), while MDAgents adapts the collaboration structure to estimated case complexity (Kim et al., 2024). Longitudinal EHR interaction adds database navigation, temporal eligibility, and delayed labels: MIRA studies autonomous agents in a sandboxed EHR environment (Ferber et al., 2026); DeepRare emphasizes traceable rare-disease reasoning (Zhao et al., 2026); MedAgentGym supports code-centric biomedical agent training (Xu et al., 2026b); and AgentEHR with RetroSum addresses EHR navigation and retrospective summarization (Liao et al., 2026). Contemporary EHR-agent systems emphasize collaborative modeling (ColaCare), executable interaction (EHRAgent), dynamic planning (DynamiCare), or confidence-aware verification (Wang et al., 2025; Shi et al., 2024; Shang et al., 2026). Their tasks, data access, and metrics differ from MedEHR, so we reimplement them under a protocol-matched setting; the corresponding comparison is reported in the Appendix E. MedMDT differs by making

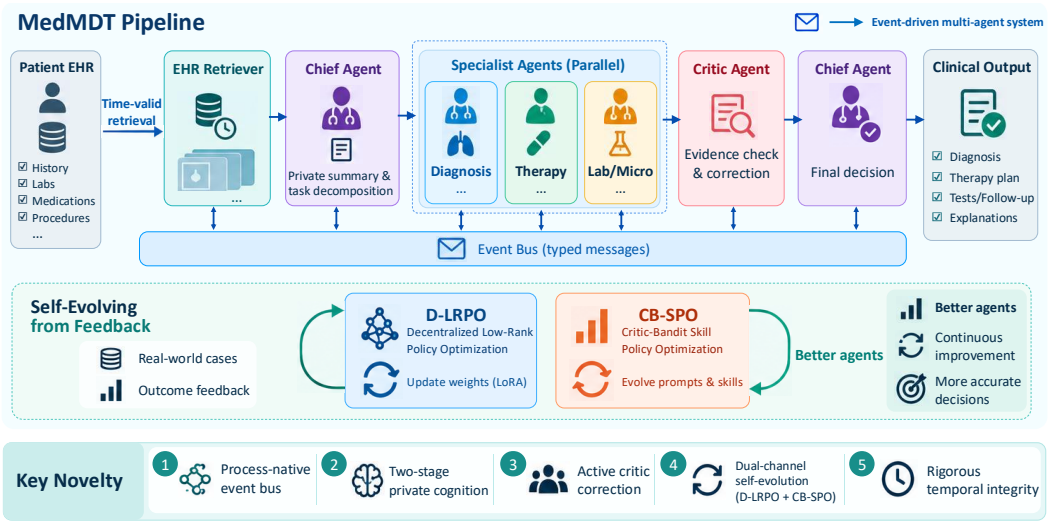


Figure 2: The overview of MedMDT (Medical multidisciplinary agent teams) pipeline. The events connect retrieval, chief coordination, task-routed specialists, critique, and final synthesis. D-LRPO updates specialist adapters; CB-SPO updates reusable instructions.

role ownership persistent across cases and coupling within-case critique to two separately controlled adaptation channels.

2.2 Online Reinforcement Learning

Policy optimization can improve a language model through scored rollouts. PPO provides a clipped policy-update objective (Schulman et al., 2017), group-relative policy optimization (GRPO) estimates relative advantages without a separately trained value model (Shao et al., 2024), and LoRA restricts updates to low-rank factors (Hu et al., 2021). A complementary line adapts external, human-readable state. Self-Refine and Reflexion use generated feedback to revise behavior (Madaan et al., 2023; Shinn et al., 2023; Li et al., 2026a); ExpeL and ReasoningBank distill experience into reusable guidance (Zhao et al., 2024; Ouyang et al., 2026); and EvoSkill and SkillOpt optimize reusable skills (Alzubi et al., 2026; Yang et al., 2026). Q-Evolve combines process-level supervision with policy learning (Zhang et al., 2026). MedMDT's contribution is not GRPO, LoRA, or bandit selection in isolation. It is their role-aware integration into a clinical team whose parametric and semantic states have separate ownership, validation, and rollback, with shared reward released only after prediction commitment. CB-SPO uses UCB-inspired prioritization (Auer et al., 2002); because rules interact and the policy changes over time, it does not inherit classical independent-arm regret guarantees.

3 MedMDT

3.1 Overview

For case i, let q_i denote the task, t_i the prediction time, $\mathcal{C}_{q_i}$ the candidate vocabulary, and $Y_i \subseteq \mathcal{C}_{q_i}$ the target set. The inference view is $x_i = (q_i, t_i, \mathcal{H}_{<t_i}, \mathcal{C}_{q_i})$, where $\mathcal{H}_{<t_i}$ contains only permitted historical observations. The output is a candidate set $\widehat{Y}_i$ with an evidence-oriented rationale. Targets remain private to scoring and training; they are removed before agent dispatch.

A fixed task router activates diagnosis, therapy, and laboratory/microbiology specialists as appropriate. Procedures activate diagnosis and therapy; all three participate in the RCT tasks. Each specialist produces a structured candidate proposal. The critic receives the admissible evidence and proposals and returns supported, unsupported, missing, and corrected candidates. The chief integrates these signals within a bounded interaction budget and normalizes the committed output to task vocabulary.

The persistent team state before case i is $\Theta_i = (\phi_{1:K,i}, \mathcal{S}_i)$, where $\phi_{k,i}$ denotes role k's adapter and $\mathcal{S}_i$ the versioned skill registry. Inference maps (x_i, Θ_i) to $\widehat{Y}_i$ without access to Y_i. Only after commitment does the adaptation workflow receive a reward and propose Θ_{i+1}. D-LRPO may update

one role-local ϕ_k; CB-SPO may update a bounded suffix in $\mathcal{S}$; neither changes the frozen backbone. During a dynamic-rule trial, competing role updates and rule reselection are deferred. This state factorization and synchronization protocol is the central distinction from an unstructured committee or a single jointly updated agent.

Distinct operating-system processes communicate through queues; role state and update targets are explicit in Figure 2. Thus "decentralized" refers to specialist policy ownership and inference, while a dedicated trainer schedules optimization. It does not imply statistically independent agents or fully decentralized optimization.

3.2 Decentralized Low-Rank Policy Optimization (D-LRPO)

Let W_0 be a frozen specialist backbone. For role k, an adapted projection is

$$W_k = W_0 + \frac{\alpha}{r} B_k A_k, \qquad A_k \in \mathbb{R}^{r \times d_{\text{in}}}, \quad B_k \in \mathbb{R}^{d_{\text{out}} \times r}. \tag{1}$$

The role's policy $\pi_{\phi_k}(\cdot \mid x, s_k)$ depends on its factors ϕ_k and textual skill s_k. Independent buffers and adapter registries prevent an update for one role from overwriting another role's parameters. This is parameter isolation, not independence of downstream team predictions. At update t, training initializes from the last accepted adapter $\phi_{k,t}$ and produces a candidate $\phi_{k,t+1}$; it does not restart from scratch. Specialist completions are scored against the full task target set Y_i, rather than role-projected targets.

For a buffered training prompt, GRPO samples G fresh completions o_g and evaluates completion rewards r_g. Group-relative advantages are

$$\widehat{A}_g = \frac{r_g - \bar{r}}{\text{std}(r_1, \ldots, r_G) + \epsilon}, \qquad \bar{r} = G^{-1} \sum_{g=1}^{G} r_g. \tag{2}$$

A representative token-averaged clipped objective is

$$J_k = \mathbb{E}\left[\frac{1}{G} \sum_g \frac{1}{|o_g|} \sum_j \left\{ \min\left(\rho_{g,j} \widehat{A}_g, \text{clip}(\rho_{g,j}, 1-\eta, 1+\eta) \widehat{A}_g \right) - \beta d_{g,j} \right\} \right], \tag{3}$$

where $\rho_{g,j}$ is the current-to-rollout token probability ratio and $d_{g,j}$ is the reference-policy KL penalty estimator. Training is delegated to the pinned GRPO implementation; clipping and loss-normalization defaults belong to that implementation. Only adapter parameters are trainable.

The operational setting uses $G = 2$. For a non-tied pair, standardization makes the two advantages equal in magnitude and opposite in sign (up to ϵ), so this setting primarily supplies pairwise preference direction. We treat group size as an optimization choice and specify $G \in \{4, 8\}$ and an unstandardized control in Appendix E.

The completion reward is distinct from the team's logged case F1. Writing f for completion F1, o for token-overlap credit, n for supported novelty, u for output uniqueness, and z for a valid nonempty parsed output, the code uses

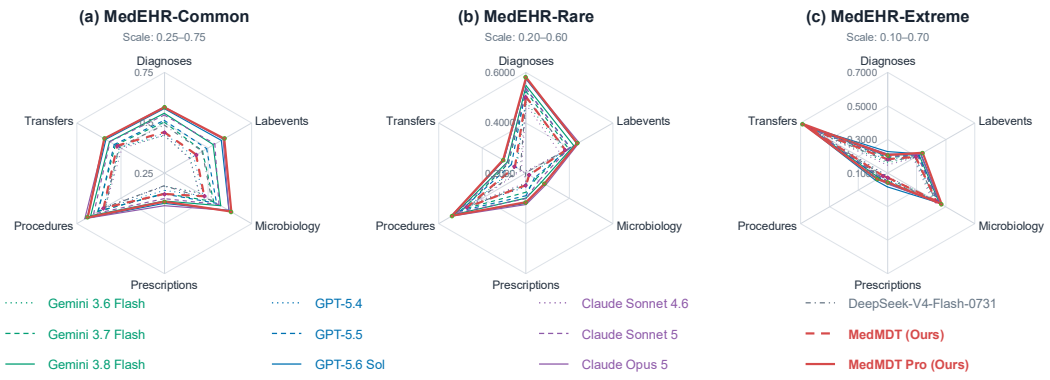
$$r = \min(1, f + (1-f)\left[0.05z + 0.25o + 0.04nu(0.25 + 0.75\max(f, o)) + 0.02u\right]). \tag{4}$$

The auxiliary terms provide graded feedback; they do not alter the reported evaluation metric. Appendix A specifies their definitions and the limits of the warm-started update mechanism.

3.3 Critic-Bandit Skill Policy Optimization (CB-SPO)

CB-SPO adapts short role- and task-specific instructions. A curated rule library covers candidate normalization, temporal evidence, missing-label review, and task constraints. For an observed rule p, selection ranks

$$U_p = \bar{R}_p + 0.05\sqrt{\frac{\log \max(N, 2)}{n_p}}, \qquad U_p = 0.25 \text{ if } n_p = 0, \tag{5}$$



MedMDT: Qwen-AgentWorld-35B-A3B + Qwen3.5-9B subagents. MedMDT Pro: DeepSeek-V4-Flash-0731 + Qwen3.8-27B subagents. Reasoning effort: medium.

Figure 3: The quantitative comparison (Sample-F1) between proposed MedMDT and SOTA LLMs.

where n_p is its observation count and N the relevant observation total. Active rules receive the shared case-level F1 reward. Rules are packed into a bounded suffix; selection is reconsidered every four rewards or when reward is below 0.35. Credit is therefore associative and shared across active rules, rather than a causal estimate of each rule's benefit.

Because an unobserved rule receives the finite priority 0.25. Therefore, exhaustive exploration is not guaranteed and a forced-coverage control is specified in Appendix E.

The dynamic channel asks the critic to propose generic rules from aggregate error summaries. These summaries contain task, reward, precision/recall/F1, and error counts, excluding raw records and target strings. Candidate rules pass sanitization and a subsequent-case validation gate. With default windows of eight baseline and eight validation cases, promotion requires an observed mean reward increase of at least 0.01. Related adapter updates and prompt reselection are deferred during the trial. This reduces concurrent-update confounding but does not remove case-mix drift or constitute a significance test. Rejected rules are rolled back; accepted suffixes are versioned.

3.4 MedEHR Benchmark

MedEHR benchmark contains 5,100 evaluation task instances and 600 training task instances from MedEHR-Common by default. MedEHR-Common and MedEHR-Rare each contain 1,200 instances (200 per task) from the corresponding MIMIC-IV partitions (Johnson et al., 2023; Liao et al., 2026). MedEHR-Extreme also contains 1,200 MIMIC-III instances (Johnson et al., 2016), covering the same six task families under dataset and schema shift.

MedEHR-RCT contains 1,500 instances (500 per task) derived from ELAIA-1, an AKI-alert randomized trial (Wilson et al., 2021; Mutter et al., 2019). At randomization (T_0), the model predicts first-24-hour care actions or adverse outcomes occurring by day 14. At T_{24}, it predicts events occurring after hour 24 and through day 14 among patients free of the primary composite event through that landmark. Outcome vocabularies include AKI progression, dialysis, death, and explicit no-event candidates. The RCT origin supplies structured timing and intervention metadata; it does not turn this prediction study into a randomized evaluation of MedMDT.

4 Experiments

4.1 Setup

MedMDT uses Qwen-AgentWorld-35B-A3B (Zuo et al., 2026) as chief and Qwen3.5-9B (Qwen Team, 2026a) for specialist roles. MedMDT Pro uses DeepSeek-V4-Flash-0731 (Xu et al., 2026a) with Qwen3.8-27B (Qwen Team, 2026b) subagents. MedMDT mini only uses Qwen-AgentWorld-35B-A3B (Zuo et al., 2026) as chief. Frontier comparisons pair each model family with its reported harness; same-chief comparisons use Qwen-AgentWorld-35B-A3B. The MIMIC-IV adaptation stream is disjoint from frozen evaluation instances. RCT evaluation uses a separate ELAIA-1 conversion, grouped by `subject_id` before sampling so that one patient cannot cross the

Table 1: Comparison of MedMDT with agentic baselines on MedEHR-Common. Performance is measured by sample-wise F1 (Sample-F1). **Bold** and underline indicate the best and second-best results, respectively. MedMDT mini involves no parameter fine-tuning and no auxiliary subagents.

Agent Model	Agent Method	Diagnoses	Labevents	Microbiology	Prescriptions	Procedures	Transfers	Average ↑
Qwen-AgentWorld-35B-A3B	MIRA (Nature 2026)	0.3689	0.3412	0.3507	0.2199	0.4673	0.3915	0.3566
Qwen-AgentWorld-35B-A3B	DeepRare (Nature 2026)	0.3784	0.3529	0.3694	0.2476	0.4739	0.4142	0.3727
Qwen-AgentWorld-35B-A3B	ReSum (arXiv 2025)	0.3766	0.3371	0.3653	0.2308	0.4714	0.3982	0.3632
Qwen-AgentWorld-35B-A3B	RetroSum (arXiv 2026)	0.4031	0.3628	0.4026	0.2529	0.5145	0.4247	0.3934
Qwen-AgentWorld-35B-A3B	ReasoningBank (ICLR 2026)	0.3932	0.3445	0.3795	0.2543	0.5096	0.4258	0.3845
Qwen-AgentWorld-35B-A3B	SkillOPT (arXiv 2026)	0.4084	0.3723	0.4056	0.2634	0.5240	0.4315	0.4009
Qwen-AgentWorld-35B-A3B	EvoSkill (COLM 2026)	0.3910	0.3482	0.3874	0.2467	0.4978	0.4106	0.3803
Qwen-AgentWorld-35B-A3B	Med-Copilot (ICLR 2026)	0.4121	0.3744	0.3967	0.2703	0.5326	0.4432	0.4049
Qwen-AgentWorld-35B-A3B	Q-Evolve (ICML 2026)	0.4103	0.3697	0.4161	0.2615	0.5532	0.4538	0.4108
Qwen-AgentWorld-35B-A3B	**MedMDT mini (Ours)**	0.4267	0.3804	0.4310	0.2843	0.5621	0.4819	0.4277
Qwen-AgentWorld-35B-A3B	**MedMDT (Ours)**	**0.4532**	**0.4328**	**0.4795**	**0.3548**	**0.5983**	**0.5194**	**0.4730**

Table 2: Comparison of MedMDT with state-of-the-art LLM agents on MedEHR-RCT. Performance is measured by sample-wise F1 (Sample-F1). **Bold** and underline indicate the best and second-best results, respectively.

Agent Model	Agent Framework	Care Actions 24h	Outcomes 14d	Residual Outcomes 24h	Average ↑
Gemini 3.6 Flash	Google Antigravity	0.2732	0.6904	0.7712	0.5783
Gemini 3.7 Flash	Google Antigravity	0.3040	0.7027	0.8143	0.6070
Gemini 3.8 Flash	Google Antigravity	0.3253	0.7181	0.8225	0.6220
GPT-5.4	Codex	0.2924	0.6326	0.7807	0.5686
GPT-5.5	Codex	0.3108	0.7119	0.8156	0.6128
GPT-5.6 Sol	Codex	0.3371	0.7405	0.8342	0.6373
Claude Sonnet 4.6	Claude Code	0.2617	0.6598	0.7509	0.5575
Claude Sonnet 5	Claude Code	0.3149	0.7192	0.8118	0.6153
Claude Opus 5	Claude Code	**0.3402**	0.7320	0.8301	0.6341
DeepSeek-V4-Flash-0731	DeepSeek Harness	0.2875	0.6773	0.7642	0.5763
Qwen-AgentWorld-35B-A3B	**MedMDT (Ours)**	0.3019	0.7148	0.8212	0.6126
DeepSeek-V4-Flash-0731	**MedMDT Pro (Ours)**	0.3346	**0.7613**	**0.8630**	**0.6530**

boundary; split manifests are versioned. MedEHR-Rare, MedEHR-Extreme and MedEHR-RCT are evaluation-only. More details are in the Appendix B.

For transformed prediction and target sets P_i and T_i, the primary metric is

$$\text{Sample-F1} = \frac{1}{n}\sum_{i=1}^{n}\frac{2|P_i \cap T_i|}{|P_i| + |T_i|}, \tag{6}$$

with empty predictions assigned zero as in the evaluator. Observational tasks have nonempty target sets; RCT absence is represented by an explicit negative/no-event label. Appendix C.1 documents laboratory-label normalization. Evaluation freezes adapters and prompt state; the workflow trains only on designated adaptation partitions and evaluates frozen, nonoverlapping manifests.

4.2 Performance on MedEHR Benchmark

Table 1, Table 2, and Figure 3 summarize the main comparisons; complete task-level results appear in Appendix D. With the same chief backbone, MedMDT achieves 0.4730 versus 0.4108 for Q-Evolve (Zhang et al., 2026), a gain of 0.0622 (15.1% relative improvement). The improvement is broad rather than driven by one endpoint: MedMDT exceeds Q-Evolve on all six Common tasks, with absolute gains from 0.0429 on Diagnoses to 0.0933 on Prescriptions. MedMDT mini attains 0.4277, indicating that the chief workflow alone is competitive with the listed methods; the full team adds 0.0453 and improves every task over mini, most strongly Prescriptions (+0.0705) and Labevents (+0.0524). This pattern is consistent with the specialists adding value where evidence is distributed across heterogeneous event streams.

Table 1 provides a protocol-matched comparison under a shared chief backbone, frozen MedEHR-Common evaluation set, identical candidate vocabularies, and the same sample-wise F1 evaluator. In particular, MedMDT mini excludes auxiliary subagents and parameter adaptation, providing a capacity-controlled comparison with the evaluated agentic baselines. The results show that MedMDT outperforms the evaluated same-chief agentic baselines.

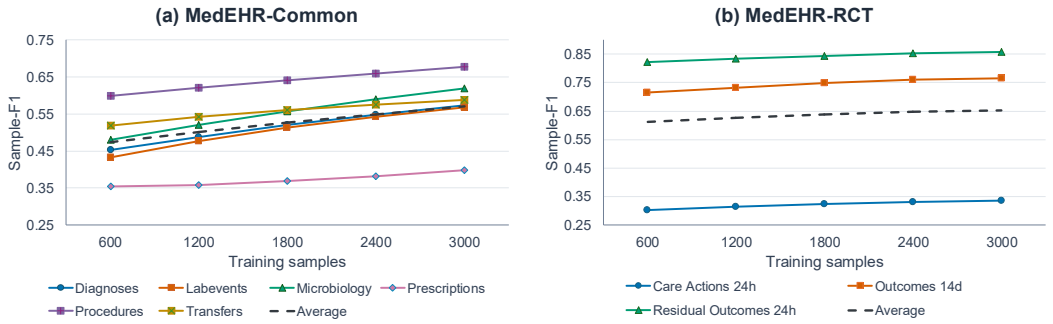


Figure 4: Online self-evolution evaluation of MedMDT on MedEHR-Common and MedEHR-RCT.

Table 3: Ablation study of the multidisciplinary agent team design on MedEHR-Common. Performance is measured by sample-wise F1 (Sample-F1). **Bold** indicates the best results.

Chief Agent	Doctor Agents	Critic Agent	Diagnoses	Labevents	Microbiology	Prescriptions	Procedures	Transfers	Average ↑
✓	×	×	0.4267	0.3804	0.4310	0.2843	0.5621	0.4819	0.4277
✓	✓	×	0.4471	0.4150	0.4643	0.3107	0.5849	0.5057	0.4546
✓	×	✓	0.4344	0.3937	0.4506	0.2938	0.5752	0.4970	0.4408
✓	✓	✓	**0.4532**	**0.4328**	**0.4795**	**0.3548**	**0.5983**	**0.5194**	**0.4730**

MedMDT Pro achieves 0.5800 on Common, 0.4103 on Rare, and 0.3382 on Extreme, demonstrating consistent improvements across all three evaluation settings. Relative to the DeepSeek harness using the same chief identifier, the absolute gains are 0.0985, 0.0731, and 0.0433, corresponding to relative improvements of approximately 20.5%, 21.7%, and 14.7%, respectively. The gains on Rare and Extreme extend the evidence beyond performance on Common: the advantage persists when the evaluation distribution changes, suggesting that the configured system's benefits are not confined to the most common cases. In particular, Rare exhibits a relative improvement comparable to that on Common, despite its lower absolute performance. The positive margin on Extreme is still consistent with partial robustness under more pronounced distribution shift. A budget-capped comparison is specified in Appendix E to further distinguish architectural contributions.

On RCT, MedMDT Pro reaches an average score of 0.6530, exceeding GPT-5.6 Sol's 0.6373 by 0.0157 and the DeepSeek harness with the same chief identifier by 0.0767. The latter corresponds to an approximately 13.3% relative improvement, extending the system-level advantage to the RCT evaluation. The task-level breakdown clarifies where this improvement arises: MedMDT Pro gains 0.0840 on Outcomes 14d and 0.0988 on Residual Outcomes 24h over the DeepSeek harness. Positive gains on both outcome tasks show that average improvement is supported by multiple endpoints rather than a single isolated result, with the larger gain occurring on Residual Outcomes 24h.

4.3 Online Self-Evolution Evaluation

Figure 4 shows the increasing reported performance over five adaptation budgets. Common rises from 0.4730 at 600 cases to 0.5699 at 3,000, a gain of 0.0969 (20.5% relative); all six tasks improve. Microbiology has the largest absolute gain (+0.1389), whereas Prescriptions remains the lowest-scoring task despite improving by 0.0431. RCT rises from 0.6126 to 0.6532, with gains of 0.0346, 0.0501, and 0.0369 across its three tasks. The successive 600-case average gains decrease from 0.0282 to 0.0208 on Common and from 0.0143 to 0.0057 on RCT, suggesting diminishing marginal returns over the observed range rather than saturation being reached. These trajectories demonstrate association with additional adaptation data, without isolating sample count from update count, case order, or training compute.

4.4 Ablation Studies

4.4.1 Ablation Study on Proposed Components

With the default adaptation mechanisms enabled, activating the specialist pathway raises Common Sample-F1 from 0.4277 for chief-only inference to 0.4546 (+0.0269), while chief-plus-critic inference reaches 0.4408 (+0.0131). Their combination reaches 0.4730 and improves every task over chief-only inference. The adapted specialist pathway yields its largest gains on Labevents (+0.0346) and Microbiology (+0.0333), where relevant evidence is distributed across measurements, specimen

Table 4: Ablation study of policy optimization on MedEHR-Common. Performance is measured by sample-wise F1 (Sample-F1). **Bold** indicates the best results.

D-LRPO	CB-SPO	Diagnoses	Labevents	Microbiology	Prescriptions	Procedures	Transfers	Average ↑
×	×	0.4234	0.3915	0.4341	0.2872	0.5579	0.4808	0.4292
✓	×	0.4453	0.4184	0.4630	0.3095	0.5836	0.5061	0.4543
×	✓	0.4301	0.3959	0.4502	0.2927	0.5714	0.4935	0.4390
✓	✓	**0.4532**	**0.4328**	**0.4795**	**0.3548**	**0.5983**	**0.5194**	**0.4730**

results, and temporally separated events. This task profile matches the intended role of specialist proposals: they expose domain-specific evidence before the chief compresses it into a single decision. The critic produces smaller but broad gains, led by Microbiology (+0.0196) and Transfers (+0.0151), consistent with a verification stage that removes unsupported candidates and identifies omissions. In the full team, the two functions compose: relative to chief-only inference, gains are largest on Prescriptions (+0.0705), followed by Labevents (+0.0524) and Microbiology (+0.0485), while Diagnoses, Procedures, and Transfers improve by 0.0265–0.0375. Thus, MedMDT adds both evidence coverage and decision checking rather than duplicating chief's reasoning (Table 3).

Without either adaptation channel, the average is 0.4292; D-LRPO alone reaches 0.4543, CB-SPO alone 0.4390, and both 0.4730. These contrasts support complementarity within the reported configuration. Appendix E specifies the factorial control in more detail.

4.4.2 Ablation Study on Chief Agent Models

With Qwen3.5-9B (Qwen Team, 2026a) subagents fixed, reported RCT averages range from 0.5541 for DeepSeek-R1-32B to 0.6486 for DeepSeek-V4-Flash-0731, while the default Qwen chief yields 0.6126. The ordering is coherent across all three endpoints: moving from the lowest- to the highest-scoring chief improves Care Actions by 0.0779, Outcomes by 0.0962, and Residual Outcomes by 0.1092. This aligned movement across endpoints shows that the chief is not only formatting specialist outputs; its capacity determines how effectively heterogeneous proposals, temporal evidence, and critique are reconciled into the final set. The 0.0945 average spread therefore identifies synthesis as a primary performance lever in the complete system (Table 5).

4.4.3 Ablation Study on Subagent Models

Holding the Qwen chief fixed, RCT averages span 0.5755 with MedGemma 1.5-4B-it to 0.6384 with Qwen-AgentWorld-35B-A3B subagents, a 0.0629 spread; Qwen3.5-9B yields 0.6126. From the lowest- to the highest-scoring subagent configuration, Care Actions, Outcomes, and Residual Outcomes improve by 0.0591, 0.0662, and 0.0634, respectively. The balanced gains indicate that stronger specialists improve both near-term action proposals and longer-horizon outcome evidence before synthesis. Together with the chief ablation, the results reveal a two-stage capacity structure: subagent quality controls proposal completeness and evidence specialization, while chief quality controls cross-proposal integration. The larger 0.0945 chief spread further highlights the value of allocating sufficient capacity to the final reconciliation stage (Table 6).

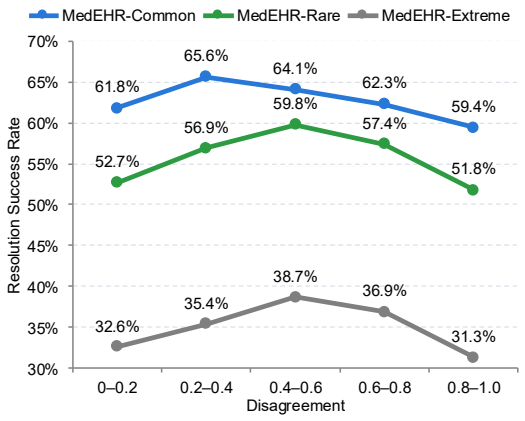


Figure 5: Resolution Success Rate (RSR) across proposal-set disagreement bins on procedures task.

4.5 Interpreting Multidisciplinary Reasoning

4.5.1 Specialist Disagreement and Resolution

Procedures activates diagnosis and therapy specialists. For their proposal sets P_{na} and P_{nb} on case n, we define disagreement as Jaccard distance, $D_n = 1 - |P_{na} \cap P_{nb}|/|P_{na} \cup P_{nb}|$. To distinguish productive reconciliation from a final answer that merely follows one specialist, we define Disagree-

Table 5: Ablation study of chief agent models on MedEHR-RCT. All settings use Qwen3.5-9B as the subagent model.

Chief Agent	Care Actions 24h	Outcomes 14d	Residual Outcomes 24h	Average
DeepSeek-V4-Flash-0731	0.3302	0.7571	0.8584	0.6486
Qwen3.8-27B	0.3231	0.7492	0.8487	0.6403
Qwen-AgentWorld-35B-A3B	0.3019	0.7148	0.8212	0.6126
Qwen3.6-27B	0.2845	0.7037	0.8026	0.5969
Gemma 4-31B-it	0.2712	0.6854	0.7751	0.5772
DeepSeek-R1-32B	0.2523	0.6609	0.7492	0.5541

Table 6: Ablation study of subagent models on MedEHR-RCT. All settings use Qwen-AgentWorld-35B-A3B as the chief agent.

Subagent	Care Actions 24h	Outcomes 14d	Residual Outcomes 24h	Average
Qwen-AgentWorld-35B-A3B	0.3204	0.7456	0.8493	0.6384
Qwen3.6-27B	0.3162	0.7320	0.8401	0.6294
Qwen3.5-9B	0.3019	0.7148	0.8212	0.6126
Gemma 4-12B	0.2963	0.7082	0.8130	0.6058
Gemma 4-E4B-it	0.2845	0.6923	0.8027	0.5932
MedGemma 1.5-4B-it	0.2613	0.6794	0.7859	0.5755

Table 7: Interpretability study of specialist disagreement and resolution on Procedures task. Performance is measured by the sample-wise F1 score (Sample-F1).

Disagreement	MedEHR-Common	MedEHR-Rare	MedEHR-Extreme
0–0.2	0.5971	0.4549	0.1243
0.2–0.4	0.6094	0.4735	0.1314
0.4–0.6	0.5957	0.4842	0.1422
0.6–0.8	0.5892	0.4697	0.1371
0.8–1.0	0.5726	0.4420	0.1125

Table 8: Prediction transition analysis on MedEHR-Common dataset.

Task	TP Preserve	FP Remove	FN Recovery
Diagnoses	95.2%	38.9%	32.4%
Labevents	93.3%	34.1%	48.8%
Microbiology	93.8%	43.7%	50.6%
Prescriptions	90.8%	55.4%	38.2%
Procedures	97.2%	50.3%	41.8%
Transfers	96.5%	60.2%	23.6%

ment Resolution Gain as $\mathrm{DRG}_n = \mathrm{F1}(P_{nf}, Y_n) - \max\{\mathrm{F1}(P_{na}, Y_n), \mathrm{F1}(P_{nb}, Y_n)\}$, where P_{nf} is the chief's final set. A case is successfully resolved when $\mathrm{DRG}_n > 0$; the Resolution Success Rate (RSR) is the proportion of successful cases within a disagreement bin. Appendix C.1 gives the full definitions and edge-case conventions. Common RSR peaks at 65.6% for disagreement 0.2–0.4; Rare and Extreme peak at 59.8% and 38.7% at 0.4–0.6 as shown in Figure 5. Final Sample-F1 follows the same nonmonotonic pattern: it peaks under mild disagreement on Common (0.6094) but under moderate disagreement on Rare (0.4842) and Extreme (0.1422), then declines in the highest bin for every subset as shown in Table 7. RSR uses the strict criterion $\mathrm{DRG} > 0$: when a specialist is already optimal, reconciliation has no headroom to count as a success even if it preserves that answer. Bin counts and a match-or-exceed-best sensitivity analysis are specified in Appendix E.

4.5.2 Prediction Transition Analysis

Let $P_n^{\mathrm{pre}} = \mathrm{Norm}(\bigcup_{k \in \mathcal{A}_n} P_{nk})$ be the normalized, deduplicated union of all active specialist proposals before reconciliation, P_n^{post} the committed final set, and Y_n the target. Correct Prediction Preservation Rate (CPPR, TP Preserve), False Positive Correction Rate (FPCR, FP Remove) and False Negative Recovery Rate (FNRR, FN Recovery) are label-level, micro-aggregated transition rates; Appendix C.1 specifies their set formulas.

The transition results on Common yield CPPR of 90.8–97.2%, FPCR of 34.1–60.2%, and FNRR of 23.6–50.6% (Table 8). Averaged equally across the six task-level rates, correction preserves 94.5% of true positives while removing 47.1% of false positives and recovering 39.2% of false negatives. This rules out a trivial "predict fewer labels" explanation: reconciliation is predominantly conservative, yet it also adds a substantial fraction of missed targets. Prescriptions shows the most aggressive precision-oriented behavior (55.4% FPCR and the lowest CPPR, 90.8%); Procedures has the highest preservation (97.2%) while correcting both error types; Transfers has the largest FPCR (60.2%) but the lowest FNRR (23.6%); and Microbiology has the largest FNRR (50.6%). And it shows that MedMDT has task-dependent correction capability rather than applying a uniform rule.

5 Conclusion

MedMDT treats a clinical agent team as an evolving, factorized policy rather than a transient group conversation. Task-routed specialists, a coordinating chief, and an evidence critic operate behind explicit temporal and label boundaries; D-LRPO updates role-private parametric state, whereas CB-SPO updates bounded, versioned semantic state. The reported comparisons favor the full team over the evaluated same-chief agentic baselines, and checkpoint performance rises with additional adaptation data. These results support the value of specialization, critique, and dual-state evolution, while cross-dataset degradation and the limitations of retrospective evaluation remain material. MedMDT therefore provides a concrete and auditable framework for studying self-improving clinical teams, not evidence for autonomous clinical deployment.

AI USE STATEMENT

We used generative AI tools to search for and summarize relevant literature and to assist with language editing. We did not use them to formulate mathematical claims, construct proofs, propose hypotheses, design or implement the method, analyze qualitative data, interpret experimental results, develop theoretical models, translate the manuscript, or clean and reformat the datasets. The authors verified all AI-assisted material and take full responsibility for the manuscript's content.

ETHICS STATEMENT

This work studies retrospective prediction using deidentified EHR benchmarks and a secondary representation of the ELAIA-1 trial; it neither assigns interventions nor evaluates deployment in patient care. The original trial's ethical oversight does not automatically authorize every secondary use. Data access, processing, and release of derived artifacts must follow the source agreements and applicable institutional review requirements; we do not assert an IRB identifier or exemption absent from the available documentation. Patient-level records and trajectories must remain in authorized environments, particularly when external inference services are considered. Aggregate error summaries reduce direct exposure to the text optimizer but provide no formal privacy guarantee. Historical care actions reflect local practice and unequal access, not necessarily optimal care, and average F1 does not establish calibration, subgroup equity, or clinical benefit. Consistent with responsible clinical-ML guidance (Wiens et al., 2019; Lekadir et al., 2025), any prospective use would require locked-model validation, human-factors evaluation, monitoring, and clinician oversight. The present results do not establish safety or effectiveness for autonomous care.

REPRODUCIBILITY STATEMENT

Sections 3–4 define the system and evaluation. Appendix A states the objectives, assumptions, and limits of the two update channels; Appendix B specifies task construction, feature horizons, and leakage controls; Appendix C documents metrics, message schemas, hyperparameters, environment, and hardware; Appendix D retains all task-level results and ablations; and Appendix G adds prompt contracts, and state-versioning requirements. Reporting is informed by TRIPOD-LLM (Gallifant et al., 2025); live clinical evaluation, which is outside this study, would additionally require stage-appropriate guidance such as DECIDE-AI (Vasey et al., 2022).

REFERENCES

Salaheddin Alzubi, Noah Provenzano, Jaydon Bingham, Weiyuan Chen, and Tu Vu. Evoskill: Automated skill discovery for multi-agent systems. *arXiv preprint arXiv:2603.02766*, 2026.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Dyke Ferber, Lars Hilgers, Christiane Höper, Benedict Kinny-Köster, Jan-Niklas Eckardt, Katharina Egger-Heidrich, Marius Bill, Martin MK Schneider, Jan Clusmann, Lejla Kadric, et al. Towards autonomous medical artificial intelligence agents. *Nature*, pp. 1–10, 2026.

Jack Gallifant, Majid Afshar, Saleem Ameen, Yindalon Aphinyanaphongs, Shan Chen, Giovanni Cacciamani, Dina Demner-Fushman, Dmitriy Dligach, Roxana Daneshjou, Chrystinne Fernandes, et al. The tripod-llm reporting guideline for studies using large language models. *Nature medicine*, 31(1):60–69, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):160035, 2016.

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

Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik S Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae W Park. Mdagents: An adaptive collaboration of llms for medical decision-making. *Advances in Neural Information Processing Systems*, 37:79410–79452, 2024.

Karim Lekadir, Alejandro F Frangi, Antonio R Porras, Ben Glocker, Celia Cintas, Curtis P Langlotz, Eva Weicken, Folkert W Asselbergs, Fred Prior, Gary S Collins, et al. Future-ai: international consensus guideline for trustworthy and deployable artificial intelligence in healthcare. *bmj*, 388, 2025.

Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, Steve Jiang, and You Zhang. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge. *Cureus*, 15(6), 2023.

Zihan Li, Diping Song, Zefeng Yang, Deming Wang, Fei Li, Xiulan Zhang, Paul E Kinahan, and Yu Qiao. Visionunite: A vision-language foundation model for ophthalmology enhanced with clinical knowledge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Zihan Li, Feiyang Liu, Dandan Shan, Ruibo Wang, and Qingqi Hong. Opera: Offline policy-guided expert routing and adaptation for universal biomedical image analysis. *arXiv preprint arXiv:2607.25108*, 2026a.

Zihan Li, Yiqing Wang, Sina Farsiu, and Paul Kinahan. Boosting medical visual understanding from multi-granular language learning. In *International Conference on Learning Representations*, volume 2026, pp. 110295–110334, 2026b.

Yusheng Liao, Chuan Xuan, Yutong Cai, Lina Yang, Zhe Chen, Yanfeng Wang, and Yu Wang. Agentehr: Advancing autonomous clinical decision-making via retrospective summarization. *arXiv preprint arXiv:2601.13918*, 2026.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Marina Mutter, Melissa Martin, Yu Yamamoto, Aditya Biswas, Boian Etropolski, Harold Feldman, Amit Garg, Noah Gourlie, Stephen Latham, Haiqun Lin, et al. Electronic alerts for acute kidney injury amelioration (elaia-1): a completely electronic, multicentre, randomised controlled trial: design and rationale. *BMJ open*, 9(5):e025117, 2019.

Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, volume 2026, pp. 94327–94354, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026b. URL https://qwen.ai/blog?id=qwen3.8.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Dandan Shan, Zihan Li, Yunxiang Li, Qingde Li, Jie Tian, and Qingqi Hong. Stpnet: Scale-aware text prompt network for medical image segmentation. *IEEE Transactions on Image Processing*, 34:3169–3180, 2025.

Tianqi Shang, Weiqing He, Charles Zheng, Lingyao Li, Li Shen, and Bingxin Zhao. Dynamicare: A dynamic multi-agent framework for interactive and open-ended medical decision-making. *AMIA Summits on Translational Science Proceedings*, 2026:408, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Jieyu Zhang, Hang Wu, Yuanda Zhu, Joyce C Ho, Carl Yang, and May Dongmei Wang. Ehragent: Code empowers large language models for few-shot complex tabular reasoning on electronic health records. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22315–22339, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Xiangru Tang, Anni Zou, Zhuosheng Zhang, Ziming Li, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. Medagents: Large language models as collaborators for zero-shot medical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 599–621, 2024.

Baptiste Vasey, Myura Nagendran, Bruce Campbell, David A Clifton, Gary S Collins, Spiros Denaxas, Alastair K Denniston, Livia Faes, Bart Geerts, Mudathir Ibrahim, et al. Reporting guideline for the early-stage clinical evaluation of decision support systems driven by artificial intelligence: Decide-ai. *Nature medicine*, 28(5):924–933, 2022.

Zixiang Wang, Yinghao Zhu, Huiya Zhao, Xiaochen Zheng, Dehao Sui, Tianlong Wang, Wen Tang, Yasha Wang, Ewen Harrison, Chengwei Pan, et al. Colacare: Enhancing electronic health record modeling through large language model-driven multi-agent collaboration. In *Proceedings of the ACM on Web Conference 2025*, pp. 2250–2261, 2025.

Jenna Wiens, Suchi Saria, Mark Sendak, Marzyeh Ghassemi, Vincent X Liu, Finale Doshi-Velez, Kenneth Jung, Katherine Heller, David Kale, Mohammed Saeed, et al. Do no harm: a roadmap for responsible machine learning for health care. *Nature medicine*, 25(9):1337–1340, 2019.

F Perry Wilson, Melissa Martin, Yu Yamamoto, Caitlin Partridge, Erica Moreira, Tanima Arora, Aditya Biswas, Harold Feldman, Amit X Garg, Jason H Greenberg, et al. Electronic health record alerts for acute kidney injury: multicenter, randomized clinical trial. *bmj*, 372, 2021.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, et al. Resum: Unlocking long-horizon search intelligence via context summarization. *arXiv preprint arXiv:2509.13313*, 2025.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026a.

Ran Xu, Yuchen Zhuang, Yishan Zhong, Yue Yu, Zifeng Wang, Xiangru Tang, Hang Wu, May Dongmei Wang, Peifeng Ruan, Donghan Yang, et al. Medagentgym: A scalable agentic training environment for code-centric reasoning in biomedical data science. In *International Conference on Learning Representations*, volume 2026, pp. 89688–89730, 2026b.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, et al. Skillopt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*, 2026.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Yudi Zhang, Meng Fang, Zhenfang Chen, and Mykola Pechenizkiy. Self-evolving llm agents with in-distribution optimization. *arXiv preprint arXiv:2606.07367*, 2026.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Weike Zhao, Chaoyi Wu, Yanjie Fan, Pengcheng Qiu, Xiaoman Zhang, Yuze Sun, Xiao Zhou, Shuju Zhang, Yu Peng, Yanfeng Wang, et al. An agentic system for rare disease diagnosis with traceable reasoning. *Nature*, 651(8106):775–784, 2026.

Yuxin Zuo, Zikai Xiao, Li Sheng, Fei Huang, Jianhong Tu, Yuxuan Liu, Tianyi Tang, Xiaomeng Hu, Yang Su, Qingfeng Lan, et al. Qwen-agentworld: Language world models for general agents. *arXiv preprint arXiv:2606.24597*, 2026.

APPENDIX

Table of Contents

A Detailed Theoretical Analysis of MedMDT **15**

A.1 System State and Information Boundaries 15

A.2 D-LRPO 15

A.3 CB-SPO 16

B Dataset Details **17**

B.1 Dataset Sources 17

B.2 Dataset Construction 17

C Setup Details **19**

C.1 Evaluation Metrics 19

C.2 Implementation Details 20

D Detailed Experimental Results **21**

D.1 Detailed Results on MedEHR-Common 21

D.2 Detailed Results on MedEHR-Rare 21

D.3 Detailed Results on MedEHR-Extreme 22

D.4 Detailed Results on Online Self-evolution Evaluation 22

E Extended Evaluation **23**

E.1 Architecture-by-Adaptation Analysis 23

E.2 Direct EHR-Agent Comparisons 25

E.3 D-LRPO Group Size and Advantage Normalization 26

E.4 CB-SPO Exploration and Rule Coverage 26

E.5 RCT Label Distribution and Positive-Class Evaluation 27

E.6 Specialist Disagreement and Chief Resolution 28

F Discussion **29**

F.1 Interpretation of the Evidence 29

F.2 Failure Modes and Limitations 30

F.3 Clinical Scope 30

G Operational Protocol **30**

G.1 Case Lifecycle 30

G.2 Protocol Invariants 31

G.3 Prompt and Output Contracts 32

A Detailed Theoretical Analysis of MedMDT

This appendix separates algebraic properties from assumptions required for learning guarantees. D-LRPO is a role-specific application of low-rank policy optimization, and CB-SPO is a heuristic text-policy optimizer with a validation gate.

A.1 System State and Information Boundaries

Let $\mathcal{F}_i^-$ contain the admissible case view x_i, the frozen pre-case team state Θ_i, and inference randomness, but not the target Y_i or any observation at or after the task-specific horizon. Let τ_i denote the complete pre-commit message transcript.

Proposition 1 (Commit-before-feedback invariant)**.** *If every inference message is generated only from $\mathcal{F}_i^-$ and supported retrieval tools enforce the same time horizon, then both τ_i and $\widehat{Y}_i$ are measurable with respect to $\mathcal{F}_i^-$. Releasing Y_i only after the final-decision event therefore prevents direct target access through the defined message protocol.*

Proof. The initial case message is a function of x_i with the target removed. Each subsequent retriever, specialist, critic, and chief message is a function of earlier messages, admissible tool output, fixed pre-case state, and inference randomness. Closure under composition gives the result. This is a protocol-level non-access statement: it does not exclude target information memorized during pretraining, copied-forward information in a historical field, or leakage through an unsupported external tool. □

The second boundary concerns adaptation. A committed prediction first produces a scored event; the event may then update a role buffer, rule statistics, or a candidate state version. During dynamic-rule validation, the relevant adapters and prompt selection are held fixed. This makes the compared baseline and candidate windows easier to audit, but it does not randomize their case mix and therefore does not identify the causal effect of a rule.

A.2 D-LRPO

Parameter isolation and size. For one $d_{\text{out}} \times d_{\text{in}}$ projection, Equation 1 introduces $r(d_{\text{in}} + d_{\text{out}})$ trainable parameters per role, compared with $d_{\text{in}} d_{\text{out}}$ for a dense update. For K roles the trainable adapter count is $Kr(d_{\text{in}} + d_{\text{out}})$. This counts adaptation parameters, not the memory consumed by separately instantiated frozen models or optimizer state.

Proposition 2 (Low-rank and role-local updates)**.** *For each adapted projection,* $\operatorname{rank}(W_k - W_0) \leq r$. *If role k's update writes only (A_k, B_k), then the parameters of every other role remain unchanged.*

Proof. The product-rank inequality gives $\operatorname{rank}(B_k A_k) \leq \min(\operatorname{rank}(B_k), \operatorname{rank}(A_k)) \leq r$. Disjoint parameter storage means that writing coordinates of ϕ_k changes no coordinate of ϕ_ℓ for $\ell \neq k$. The chief's eventual output can still change because it depends on role k's proposal. □

A conditional behavioral bound. For a fixed input and fixed other agents, let P and Q be the old and new distributions of a specialist's complete proposal, and let $g(o) \in [0, 1]$ be the expected final utility conditioned on that proposal and a fixed downstream mechanism. If $D_{\text{KL}}(Q\|P) \leq \delta$, then

$$|\mathbb{E}_Q g - \mathbb{E}_P g| \leq \text{TV}(Q, P) \leq \sqrt{\delta/2}. \tag{7}$$

The first inequality follows from the variational characterization of total variation; the second is Pinsker's inequality. This is a bound under a *sequence-distribution KL assumption*. A token-level reference penalty, adapter rank, and clipped surrogate do not establish that assumption in a realized run. In particular, this inequality guarantees neither improvement nor clinical safety.

Relative advantage and reward. If all completions in a group receive identical reward, the centered advantages are zero (up to numerical handling). Thus small groups can supply weak information on uniformly solved or uniformly failed examples. The default $G = 2$ makes the reward shaping relevant, but its independent contribution has not been ablated. In Equation 4, f is computed by the

same task-aware set scorer used elsewhere. The overlap term o averages, over predicted labels, their best token-set Jaccard similarity to any gold label after the reward tokenizer's stop-word removal. The uniqueness term is the number of distinct lowercased prediction strings divided by the number of predictions. Novelty n is the fraction of unique predictions that occur in the permitted candidate hints but not in the current system prediction. If those hints cannot be recovered from the prompt, novelty is zero. The parser indicator z requires both a parsed object and nonempty predictions. Exact $f = 1$ receives reward 1 directly. For $f, o, n, u, z \in [0, 1]$, the shaped addition is at most $0.36(1 - f)$, so the reward remains bounded in $[0, 1]$. Auxiliary credit can reward partially incorrect outputs and should not be interpreted as clinical utility.

Implemented update semantics. The trainer buffers role-relevant training cases, builds training prompts without exposing gold labels to the model, and passes targets separately to the completion reward function. Buffered case F1 is retained for logging and fallback adapter selection. The supplied `_train_grpo_lora` reuses the base model plus LoRA configuration for each update and explicitly reload the preceding role adapter as the training initialization. Accordingly, this snapshot implements repeated online fitting and selection of role-local adapters. Inference hot-loads an accepted adapter. A successful candidate is compared with the saved best selection score, using the mean training completion reward when available and buffered mean reward otherwise. Because different candidate updates may be scored on different buffers, this comparison is not a held-out guarantee of monotonic generalization. Invalid or incomplete adapter directories are not registered; the implementation can prune superseded versions. Full archival reproducibility therefore requires preserving manifests and accepted checkpoints outside the pruning lifecycle.

A.3 CB-SPO

What the bandit score means. Equation 5 is inspired by upper-confidence selection (Auer et al., 2002), but differs from classical UCB. Unobserved rules have a finite score of 0.25, so they are not guaranteed to be tried. Several rules can be active simultaneously, receive the same case reward, and influence one another. Task mix and specialist policies can change the reward distribution. As a result, independent stationary-arm regret bounds do not apply. The practical interpretation is exploration-biased prioritization of short reusable instructions under a character budget.

Validation gate and a conditional error bound. Suppose, only for analysis, a fixed candidate and a baseline are evaluated using independent bounded rewards from the same case distribution, with m_1 candidate and m_0 baseline samples. Write their means as $\widehat{\mu}_1, \widehat{\mu}_0$ and their population means as μ_1, μ_0. If the candidate does not improve the population mean, then a Hoeffding bound gives

$$\Pr(\widehat{\mu}_1 - \widehat{\mu}_0 \geq \gamma) \leq \exp\left[-\frac{2\gamma^2}{1/m_1 + 1/m_0}\right] \quad \text{when } \mu_1 \leq \mu_0. \tag{8}$$

To obtain this expression, apply Hoeffding's inequality to the weighted sum of m_1 candidate rewards with weight $1/m_1$ and m_0 baseline rewards with weight $-1/m_0$. The sum of squared ranges is $1/m_1 + 1/m_0$. An upper bound of δ would require $\gamma \geq \sqrt{\frac{1}{2}(1/m_1 + 1/m_0)\log(1/\delta)}$. The default $m_0 = m_1 = 8$, $\gamma = 0.01$ does not provide such a conventional confidence guarantee. Moreover, the implemented sequential windows need not satisfy identical case distributions, and candidate selection can depend on preceding errors. The actual gate is an empirical acceptance heuristic.

Dynamic-rule lifecycle. The optimizer aggregates errors over a task/role window and requests at most two candidate rules, each capped at 500 characters. The critic receives aggregate counts and a generic failure-mode description rather than raw patient records. Sanitization restricts the generated suffix before testing. A baseline phase leaves the candidate inactive; a candidate phase applies the trial suffix to subsequent cases. Prompt reselection is suspended while the dynamic trial is pending or active; relevant LoRA updates are deferred. On acceptance, the rule is merged with the static suffix under the character budget and versioned; on rejection, the static suffix is restored. These controls improve traceability but cannot eliminate memorization risks elsewhere in model training or certify semantic correctness of every generated rule.

End-to-end protocol.

1. Build a case view from time-valid observations and the task's candidate vocabulary; keep the gold target outside all inference messages.
2. Retrieve evidence, retain the chief's private summary, and dispatch to the fixed task-relevant specialist roles.
3. Collect proposals, request structured critique, synthesize the final candidate set, and commit the prediction before using its target for feedback.
4. During adaptation only, release eligible feedback. For RCT cases, respect the logical outcome horizon before releasing reward events.
5. Update rule statistics; if a dynamic trial is active, hold competing updates fixed. Otherwise schedule role-local adapter fitting and bounded rule reselection as configured.
6. Validate and register accepted adapter or prompt state. During evaluation, omit the trainer and keep all such state fixed; score predictions only after inference.

B Dataset Details

B.1 Dataset Sources

MIMIC-III and MIMIC-IV are deidentified clinical resources (Johnson et al., 2016; 2023). This study uses task instances organized in the supplied AgentEHR-derived benchmark directories, rather than proposing a new collection of MIMIC patient records. Each reported subset has 200 instances for each of Diagnoses, Labevents, Microbiology, Prescriptions, Procedures, and Transfers. ELAIA-1 evaluated AKI alerts across six hospitals (Wilson et al., 2021; Mutter et al., 2019). The summary describes a source table with 92 variables. One patient can contribute multiple task instances at different prediction times.

B.2 Dataset Construction

Case schema and candidate spaces. The common representation contains `case_id`, `subject_id`, `prediction_time`, `task`, and a label list, with dataset-specific metadata. The label field is private to scoring/training. Candidates are task vocabularies, not per-patient gold labels. Patient SQLite databases provide MIMIC evidence; ELAIA rows are converted into horizon-specific clinical sections and explicit candidate tables. Generic drug labels are preferred over brand, dose, route, or packaging descriptions when mapping prescriptions.

Table 9: Evaluated MedEHR tasks. Counts are task instances; patient overlap across tasks is permitted within a split. Default 600 training instances, expandable to 3,000 from MedEHR-Common.

Subset	Source and task scope	Tasks	Instances
Common	MIMIC-IV common partition; six EHR tasks	6	1,200 (Evaluation), 600 (Training)
Rare	MIMIC-IV rare partition; six EHR tasks	6	1,200 (Evaluation)
Extreme	MIMIC-III partition; six EHR tasks	6	1,200 (Evaluation)
RCT	ELAIA-1; care actions, outcomes, residual outcomes	3	1,500 (Evaluation)

MIMIC temporal access. Timed rows are filtered strictly before the prediction time, with unknown times excluded. Detail records are accessible only through supported parent joins. The system skips unsafe summary tables such as DRG tables by default. This mechanism prevents direct access through the supported query paths; it does not prove that every historical note is free of copied-forward information, that every timestamp denotes actual availability, or that a pretrained model has never encountered benchmark-derived text. All auxiliary retrieval and post-processing must obey the same horizon.

RCT task definitions. The implementation evaluates three multilabel tasks:

1. `aki_care_actions_24h` (Care Actions 24h): at randomization (T_0), predict the recorded first-24-hour action set. These are observed care behaviors, not normative recommendations.

2. `aki_outcomes_14d` (Outcomes 14d): at T_0, predict AKI progression, dialysis, and/or death within 14 days. If all are absent, assign only the explicit no-adverse-outcome candidate.

3. `aki_residual_outcomes_24h` (Residual Outcomes 24h): at $T_0 + 24$ hours, predict those outcomes occurring after the landmark and by day 14. The primary implemented eligibility rule is no composite event through day 1: `composite_outcome=0` or `time_to_composite_outcome>1`. Positive event times must be consistent with the 14-day horizon.

The residual task uses a shared composite-event-free cohort, rather than separate event-specific risk sets. The design summary reports 5,492 composite-event-free patients with 737 subsequent composite positives; it separately lists event-specific risk sets, which are not interchangeable with the implemented shared cohort. The full-cohort design counts are not substituted for the sampled 500-case evaluation denominators.

Table 10: Implemented care-action mapping. All positive fields are label-only at T_0. The explicit no-action candidate is used only when all 12 positive actions are absent.

Source field	Candidate meaning
`acearbreninpost24`	ACE/ARB/renin inhibitor administration
`aminopost24`	Aminoglycoside administration
`any_diureticpost24`	Any diuretic administration
`contrastirpost24`	Contrast-related procedure
`fluid_boluspost24`	Fluid bolus
`ivpost24`	IV fluid
`loopdiureticpost24`	Loop diuretic
`lrivpost24`	Lactated Ringer's
`nsaidpost24`	NSAID administration
`nsivpost24`	Normal saline
`uapost24`	Urinalysis
`uoppost24`	Urine output measurement

The design slides also mention repeat serum creatinine as an example action, but the supplied implementation does not include that action in `CARE_ACTION_FIELDS`. The manuscript follows the implemented 12-positive-action vocabulary. Overlapping actions, such as any diuretic and loop diuretic, remain distinct source-derived candidates; their hierarchical dependence affects interpretation of multilabel metrics.

Feature whitelist. The base T_0 view includes age and its censoring flag, sex, race, ethnicity, admission/location indicators, baseline creatinine, initial eGFR, minimum creatinine within the specified prior window, creatinine at randomization, timing-to-randomization fields, and selected laboratory measurements, vital signs, and pre-randomization exposures. Specifically, laboratory/vital fields include anion gap, bicarbonate, BNP flag, BUN, chloride, hemoglobin, platelet count, potassium, sodium, white-cell count, diastolic/systolic pressure, pulse, and respiratory rate at randomization. Exposure fields cover selected ACE/ARB/renin inhibitor, aminoglycoside, NSAID, contrast, cardiothoracic surgery, and PPI history. Field names and windows are defined explicitly in `T0_FEATURES`; broader comorbidity or severity variables mentioned in design slides are not automatically admitted.

At T_{24}, the view additionally permits the 12 first-day action fields, `delta_o2`, `delta_resp`, `diastolic_post_24_min`, `resp_post_24_max`, `spo2_post_24_min`, and `systolic_post_24_min`. Optional site and randomized-alert fields are hidden by default and exposed only through explicit flags. Their visibility must be recorded because it changes whether the task is prognosis-only or intervention-aware. Final outcomes, event times, consultation outcomes, maximum AKI stage, post-48-hour observations, and ambiguous downstream documentation are excluded from prediction input. Patient identifiers and trial-relative timing may be retained for splitting or scheduling but are not clinical features. A whitelist audit checks for future or label fields entering context.

C Setup Details

C.1 Evaluation Metrics

Sample-F1 is an arithmetic mean of per-instance set F1, not label-macro F1 and not pooled micro-F1. With equally sized task subsets, the equal-task average equals the pooled case mean before rounding. The documented table averages are retained at the supplied precision; rounding of constituent entries can change a recomputed last decimal. Empty predictions receive zero, including the empty-target case in the provided scorer. For RCT tasks, an absence of positive events is encoded by an explicit negative label, rather than an empty target set.

Task-aware normalization. For non-laboratory tasks, the evaluator lowercases nonempty strings and deduplicates them as a set. Prescription target extraction prioritizes `atc_name`, then candidate/name. For laboratory events, the implementation includes *both* each lowercased raw name and its suffix-stripped form in the evaluated set; suffixes can denote specimen or measurement method. Thus a label can contribute two strings. For example, a prediction of "Sodium, Whole Blood" and a target of "Sodium" give transformed sets of sizes two and one with one intersection, hence F1 $= 2/3$, rather than the score of 1 that a single canonicalized label would produce.

Transition and disagreement metrics. For the Procedures analysis, let P_{na} and P_{nb} be the diagnosis and therapy specialist proposal sets for case n, P_{nf} the final chief prediction, and Y_n the target. Proposal disagreement is the Jaccard distance

$$D_n = 1 - \frac{|P_{na} \cap P_{nb}|}{|P_{na} \cup P_{nb}|}, \tag{9}$$

$$f_{na} = \mathrm{F1}(P_{na}, Y_n), \quad f_{nb} = \mathrm{F1}(P_{nb}, Y_n), \quad f_{nf} = \mathrm{F1}(P_{nf}, Y_n), \tag{10}$$

$$\mathrm{DRG}_n = f_{nf} - \max\{f_{na}, f_{nb}\}. \tag{11}$$

Thus $D_n = 0$ denotes identical proposals and $D_n = 1$ denotes no overlap. If both proposals are empty, we set $D_n = 0$ by agreement convention. Positive DRG means that reconciliation outperforms the better individual specialist; zero means that it does not improve on that specialist; negative DRG means that reconciliation degrades it. For disagreement bin B with index set $\mathcal{I}_B = \{n : D_n \in B\}$,

$$\mathrm{RSR}(B) = \frac{1}{|\mathcal{I}_B|} \sum_{n \in \mathcal{I}_B} \mathbf{1}[\mathrm{DRG}_n > 0], \tag{12}$$

$$\overline{\mathrm{F1}}(B) = \frac{1}{|\mathcal{I}_B|} \sum_{n \in \mathcal{I}_B} f_{nf}. \tag{13}$$

The reported bins are $[0, 0.2), [0.2, 0.4), [0.4, 0.6), [0.6, 0.8)$, and $[0.8, 1]$. Table 7 reports $\overline{\mathrm{F1}}(B)$, whereas Figure 5 reports $\mathrm{RSR}(B)$. RSR measures the frequency, not the magnitude, of improvement and should be interpreted together with binned final F1 and bin counts.

For prediction-transition analysis, let P_n^{pre} be the logged candidate set immediately before critic/chief reconciliation and P_n^{post} the committed final set. The reported rates pool label transitions over cases:

$$\mathrm{CPPR} = \frac{\sum_n |P_n^{\mathrm{pre}} \cap P_n^{\mathrm{post}} \cap Y_n|}{\sum_n |P_n^{\mathrm{pre}} \cap Y_n|}, \tag{14}$$

$$\mathrm{FPCR} = \frac{\sum_n |(P_n^{\mathrm{pre}} \setminus Y_n) \setminus P_n^{\mathrm{post}}|}{\sum_n |P_n^{\mathrm{pre}} \setminus Y_n|}, \tag{15}$$

$$\mathrm{FNRR} = \frac{\sum_n |(Y_n \setminus P_n^{\mathrm{pre}}) \cap P_n^{\mathrm{post}}|}{\sum_n |Y_n \setminus P_n^{\mathrm{pre}}|}. \tag{16}$$

CPPR quantifies preservation of initially correct labels, FPCR quantifies removal of initially incorrect labels, and FNRR quantifies recovery of initially missed labels. All lie in $[0, 1]$ when their denominators are nonzero; a zero denominator makes the corresponding rate undefined rather than zero. The reported task groups all have nonzero denominators.

C.2 Implementation Details

Architecture and message schema. The code uses Python multiprocessing with the `spawn` start method. Each agent owns an inbox; the orchestrator routes a shared outbox. Events contain type, sender, receiver, case identifier, payload, correlation identifier, and creation time. The current enum includes 13 types: case start, EHR context, agent proposal, critique, chief decision, reward, LoRA update, prompt update, prompt-optimization request/response, case done, shutdown, and error. A process boundary makes update ownership explicit but does not by itself provide a security sandbox.

Table 11: Task router in the supplied implementation. Chief, retriever, and critic participate outside these specialist assignments.

Task	Active specialist roles
Diagnoses; Transfers	Diagnosis
Prescriptions	Therapy
Labevents; Microbiology	Laboratory/microbiology
Procedures	Diagnosis and therapy
All three ELAIA tasks	Diagnosis, therapy, and laboratory/microbiology

Prompts and output contract. Role prompts require candidate labels rather than arbitrary prose, explicitly time-valid evidence, and task-specific normalization. The chief's private summary schema includes patient state, time-valid evidence, candidate space, laboratory/microbiology signals, medication signals, uncertainty, and decision implications. Specialist output contains predictions, rationale, and confidence. Critique includes supported, unsupported, corrected, and missing-supported candidates plus concerns. Confidence is a model-generated field, not an established calibrated probability. The chief's final decision is normalized and scored as a candidate set; rationale quality is not the primary endpoint. Optional LLM-based retriever/chief summaries are disabled in `RuntimeConfig` defaults.

Table 12: The default setting for effective hyperparameters in MedMDT pipeline.

Component	Setting
LoRA	Rank 16; alpha 32; dropout 0.05; no bias training
Target projections	q, k, v, o, gate, up, down; vision/audio towers excluded
GRPO	Learning rate 10^{-5}; batch size 2; accumulation 4; 2 generations
GRPO regularization	KL coefficient 0.04; one epoch per buffered update
Training text caps	Prompt 4,096 tokens; completion 384 tokens
Adapter update trigger	Four buffered role-relevant rewards
Static text policy	Every four rewards or reward < 0.35; budget 3,000 characters
Dynamic text policy	Error window 8; at most 2 candidates; 500 characters each
Dynamic validation	8-case windows; minimum observed gain 0.01
Chief inference	Input cap 65,536; output cap 3,072; temperature 0.1
Specialist inference	Input cap 32,768; output cap 1,024; temperature 0.2
Critic inference	Input cap 32,768; output cap 768; temperature 0.2
Interaction	Maximum rounds 3; case timeout 900 seconds
ELAIA converter	Seed 2026; intervention/site hidden by default

Environment and hardware. The README specifies Python 3.11 and a CUDA 13.0-compatible environment. The supplied requirements pin PyTorch 2.12.1, Transformers 5.12.1, PEFT 0.19.1, TRL 1.7.0, Accelerate 1.14.0, and Datasets 5.0.0, among other dependencies. These are reproduced as declared environment requirements. We are using the eight-B200 setup. The real workflow script assigns GPUs 0–2 to the chief, 3/4/5 to specialists, 6 to the critic, and 7 to the trainer, with deterministic retrieval by default. Dataclass GPU defaults differ from the script; the script's command-line configuration takes precedence.

Table 13: Experimental results of MedMDT and state-of-the-art LLM agents on MedEHR-Common. MedMDT and MedMDT Pro employ Qwen3.5-9B and Qwen3.8-27B as their respective subagent models. Reasoning effort is set to medium across all models. Performance is measured by sample-wise F1 (Sample-F1). **Bold** and underline indicate the best and second-best results, respectively.

Agent Model	Agent Method	Diagnoses	Labevents	Microbiology	Prescriptions	Procedures	Transfers	Average ↑
Gemini 3.6 Flash	Google Antigravity	0.4356	0.3979	0.4483	0.3145	0.5918	0.5019	0.4483
Gemini 3.7 Flash	Google Antigravity	0.5034	0.4701	0.5316	0.3512	0.6325	0.5303	0.5032
Gemini 3.8 Flash	Google Antigravity	0.5479	0.5310	0.5742	0.3897	0.6726	0.5645	0.5467
GPT-5.4	Codex	0.4463	0.4186	0.4737	0.3365	0.6014	0.5052	0.4636
GPT-5.5	Codex	0.5126	0.4934	0.5499	0.3538	0.6280	0.5361	0.5123
GPT-5.6 Sol	Codex	0.5693	0.5758	0.6213	0.4016	0.6941	0.5837	0.5743
Claude Sonnet 4.6	Claude Code	0.4375	0.3947	0.4521	0.3168	0.5872	0.4944	0.4471
Claude Sonnet 5	Claude Code	0.5367	0.5395	0.5646	0.3770	0.6559	0.5618	0.5393
Claude Opus 5	Claude Code	**0.5802**	0.5863	0.6196	**0.4132**	**0.7045**	0.5907	**0.5824**
DeepSeek-V4-Flash-0731	DeepSeek Harness	0.4895	0.4673	0.5227	0.3119	0.5890	0.5085	0.4815
Qwen-AgentWorld-35B-A3B	**MedMDT (Ours)**	0.4532	0.4328	0.4795	0.3548	0.5983	0.5194	0.4730
DeepSeek-V4-Flash-0731	**MedMDT Pro (Ours)**	0.5761	**0.5945**	**0.6341**	0.3927	0.6892	**0.5935**	0.5800

Table 14: Experimental results of MedMDT and state-of-the-art LLM agents on MedEHR-Rare. MedMDT and MedMDT Pro employ Qwen3.5-9B and Qwen3.8-27B as their respective subagent models. Reasoning effort is set to medium across all models. Performance is measured by sample-wise F1 (Sample-F1). **Bold** and underline indicate the best and second-best results, respectively.

Agent Model	Agent Method	Diagnoses	Labevents	Microbiology	Prescriptions	Procedures	Transfers	Average ↑
Gemini 3.6 Flash	Google Antigravity	0.4824	0.3758	0.2120	0.2434	0.4539	0.2486	0.3360
Gemini 3.7 Flash	Google Antigravity	0.5229	0.3945	0.2459	0.2762	0.4875	0.2793	0.3677
Gemini 3.8 Flash	Google Antigravity	0.5463	0.4198	0.2715	0.2990	0.5286	0.2857	0.3918
GPT-5.4	Codex	0.5037	0.3826	0.2167	0.2558	0.4630	0.2427	0.3441
GPT-5.5	Codex	0.5363	0.4026	0.2374	0.2895	0.5023	0.2660	0.3724
GPT-5.6 Sol	Codex	0.5725	0.4342	0.2839	0.3207	0.5354	0.3019	0.4081
Claude Sonnet 4.6	Claude Code	0.4716	0.3585	0.1905	0.2396	0.4427	0.2203	0.3205
Claude Sonnet 5	Claude Code	0.5350	0.4017	0.2604	0.3021	0.5158	0.2833	0.3831
Claude Opus 5	Claude Code	0.5751	**0.4429**	**0.2923**	**0.3240**	**0.5446**	0.3014	**0.4134**
DeepSeek-V4-Flash-0731	DeepSeek Harness	0.4981	0.3854	0.1978	0.2479	0.4683	0.2258	0.3372
Qwen-AgentWorld-35B-A3B	**MedMDT (Ours)**	0.5022	0.3806	0.2147	0.2489	0.4718	0.2515	0.3450
DeepSeek-V4-Flash-0731	**MedMDT Pro (Ours)**	**0.5806**	0.4368	0.2852	0.3158	0.5392	**0.3040**	0.4103

D Detailed Experimental Results

The following tables preserve all supplied task-level values. Bold denotes the highest supplied value in a column and underline indicates the second-highest.

D.1 Detailed Results on MedEHR-Common

Interpretation. Table 13 shows that MedMDT Pro reaches 0.5800 average Sample-F1, only 0.0024 below the highest listed average, while attaining the best results on Labevents (0.5945), Microbiology (0.6341), and Transfers (0.5935). Against the same DeepSeek-V4-Flash-0731 backbone in the DeepSeek Harness, the structured MedMDT workflow improves the average by 0.0985 and every task by 0.0808–0.1272. The largest same-backbone gains occur on Labevents and Microbiology, which require consolidating heterogeneous observations and resolving label variants. These results locate the main benefit of MedMDT in task-structured evidence processing and reconciliation: the framework turns the same chief model into a stronger predictor across all six task types, with especially large returns when evidence is distributed across clinical event streams.

D.2 Detailed Results on MedEHR-Rare

Interpretation. On MedEHR-Rare, MedMDT Pro achieves 0.4103 average Sample-F1, ranking second by 0.0031 while producing the best Diagnoses (0.5806) and Transfers (0.3040) scores and the second-best on Labevents/Microbiology/Procedures, third on Prescriptions. Relative to the same DeepSeek chief in the baseline harness, the average rises by 0.0731, with task-level improvements from 0.0514 on Labevents to 0.0874 on Microbiology. The consistent gain across all six columns

Table 15: Experimental results of MedMDT and state-of-the-art LLM agents on MedEHR-Extreme. MedMDT and MedMDT Pro employ Qwen3.5-9B and Qwen3.8-27B as their respective subagent models. Reasoning effort is set to medium across all models. Performance is measured by sample-wise F1 (Sample-F1). **Bold** and underline indicate the best and second-best results, respectively.

Agent Model	Agent Method	Diagnoses	Labevents	Microbiology	Prescriptions	Procedures	Transfers	Average ↑
Gemini 3.6 Flash	Google Antigravity	0.1618	0.2831	0.4146	0.1220	0.1203	0.6036	0.2842
Gemini 3.7 Flash	Google Antigravity	0.1940	0.3084	0.4413	0.1474	0.1439	0.6254	0.3101
Gemini 3.8 Flash	Google Antigravity	0.2097	0.3158	0.4563	0.1611	0.1592	0.6339	0.3227
GPT-5.4	Codex	0.1802	0.2913	0.4274	0.1405	0.1439	0.6147	0.2997
GPT-5.5	Codex	0.2013	0.3128	0.4465	0.1571	0.1558	0.6426	0.3194
GPT-5.6 Sol	Codex	**0.2289**	0.3312	0.4623	**0.1826**	**0.1839**	0.6591	**0.3413**
Claude Sonnet 4.6	Claude Code	0.1474	0.2760	0.4032	0.1176	0.1152	0.5992	0.2764
Claude Sonnet 5	Claude Code	0.1991	0.3069	0.4527	0.1512	0.1486	0.6234	0.3137
Claude Opus 5	Claude Code	0.2146	0.3271	0.4638	0.1643	0.1707	0.6485	0.3315
DeepSeek-V4-Flash-0731	DeepSeek Harness	0.1735	0.2876	0.4370	0.1294	0.1311	0.6108	0.2949
Qwen-AgentWorld-35B-A3B	**MedMDT (Ours)**	0.1827	0.2949	0.4335	0.1279	0.1325	0.6310	0.3004
DeepSeek-V4-Flash-0731	**MedMDT Pro (Ours)**	0.2054	**0.3408**	**0.4714**	0.1556	0.1704	**0.6853**	0.3382

Table 16: Online self-evolution evaluation of MedMDT on MedEHR-Common. All settings employ Qwen-AgentWorld-35B-A3B as the agent model.

Training	Diagnoses	Labevents	Microbiology	Prescriptions	Procedures	Transfers	Average ↑
600 samples	0.4532	0.4328	0.4795	0.3548	0.5983	0.5194	0.4730
1200 samples	0.4874	0.4769	0.5203	0.3587	0.6212	0.5427	0.5012
1800 samples	0.5197	0.5133	0.5571	0.3690	0.6408	0.5611	0.5268
2400 samples	0.5470	0.5414	0.5892	0.3814	0.6597	0.5758	0.5491
3000 samples	0.5725	0.5672	0.6184	0.3979	0.6764	0.5872	0.5699

is important for the rare-event regime: specialist routing improves coverage of infrequent evidence patterns, and the critic/chief stages consolidate these proposals without concentrating the benefit in a single endpoint. The particularly strong Microbiology and Diagnoses improvements indicate that the multidisciplinary decomposition is effective when labels depend on sparse, domain-specific signals.

D.3 Detailed Results on MedEHR-Extreme

Interpretation. Table 15 presents the strongest distribution-shift setting. MedMDT Pro obtains 0.3382 average Sample-F1, 0.0031 below the leading listed average, and records the best Labevents (0.3408), Microbiology (0.4714), and Transfers (0.6853) scores. Using the same DeepSeek chief, MedMDT Pro improves over the baseline harness by 0.0433 on average and by 0.0262–0.0745 across every task; Transfers shows the largest absolute gain. The cross-task improvement under schema and dataset shift suggests that explicit routing and semantic rules provide reusable structure beyond the chief model's default prompting behavior. In particular, the strong laboratory, microbiology, and transfer results show that normalized task interfaces and specialist-specific evidence selection remain useful when surface forms and event distributions change.

D.4 Detailed Results on Online Self-evolution Evaluation

Interpretation. Table 16 shows a Common learning trajectory that is monotonic at both the aggregate and task levels. Common rises from 0.4730 at 600 cases to 0.5699 at 3,000, a gain of 0.0969 (20.5% relative). All six tasks improve from the first to the final checkpoint. Microbiology shows the largest endpoint gain (+0.1389), whereas Prescriptions remains the lowest-scoring task and improves by +0.0431 overall.

Interpretation. Table 17 shows that the RCT trajectory also improves at every checkpoint: average Sample-F1 rises from 0.6126 to 0.6532 as the adaptation stream grows from 600 to 3,000 samples. Outcomes at 14 days gains the most (+0.0501), followed by Residual Outcomes (+0.0369) and Care Actions (+0.0346). This ordering is meaningful because longer-horizon outcomes require

Table 17: Online self-evolution evaluation of MedMDT on MedEHR-RCT. All settings employ Qwen-AgentWorld-35B-A3B as the agent model.

Training	Care Actions 24h	Outcomes 14d	Residual Outcomes 24h	Average ↑
600 samples	0.3019	0.7148	0.8212	0.6126
1200 samples	0.3145	0.7323	0.8340	0.6269
1800 samples	0.3237	0.7485	0.8436	0.6386
2400 samples	0.3306	0.7597	0.8523	0.6475
3000 samples	0.3365	0.7649	0.8581	0.6532

integrating more temporally dispersed evidence, leaving greater scope for role-specific adaptation and reusable correction rules. Residual Outcomes starts from a high score of 0.8212 and still reaches 0.8581, showing that the dual-state updates refine already strong predictions as well as lower-scoring tasks. Together with the Common trajectory, the table demonstrates that the adaptation mechanism transfers across both multi-label EHR event prediction and landmarked trial-derived endpoints.

E Extended Evaluation

The extended experiments use three independent runs per configuration, with 600 adaptation cases per run unless stated otherwise. The default system uses Qwen-AgentWorld-35B-A3B as the chief and Qwen3.5-9B as the specialist backbone. Each trained state is frozen before evaluation. Common experiments use the Common adaptation partition; RCT adaptation uses a separate ELAIA-1 training partition with subject-level separation from evaluation. Rare and Extreme are evaluated without adaptation on those subsets.

Performance values in this appendix are arithmetic means across the three runs; values following $\pm$ are sample standard deviations across runs. Comparisons and interaction effects below are computed within the extended evaluation, without mixing single-run and repeated-run results.

The reported 95% intervals and p-values concern paired evaluation contrasts conditional on the trained states; they are distinct from the across-run standard deviations and do not represent confidence intervals based on three run-level observations. Tests are two-sided. Holm correction is applied separately to the three direct comparisons, the three full-team adaptation-versus-frozen comparisons, the five optimizer comparisons, and the two rule-exploration comparisons. Bonferroni correction is applied to the three interaction tests. Intervals are pointwise, rather than simultaneous. Dataset counts describe the fixed evaluation cohorts and are not multiplied by the number of runs.

E.1 Architecture-by-Adaptation Analysis

The configuration matrix separates specialist participation, inference-time critique, parametric adaptation, and textual-policy adaptation. All configurations use the same chief, evaluation inputs, candidate vocabulary, retrieval interface, and decoding settings. The evaluation state is frozen before scoring.

The critic indicator C controls inference-time evidence critique. When CB-SPO is enabled, its adaptation-time rule-proposal service remains available even if $C = 0$. This distinction makes the critic-free CB-SPO configurations well defined.

Specialist participation and adaptation. Frozen specialists approximately match chief-only inference (0.4260 versus 0.4277), with no clear improvement from adding unadapted proposals in this comparison. Without inference-time critique, D-LRPO raises Sample-F1 by 0.0190, CB-SPO adds 0.0040, and their combination adds 0.0286. This ordering associates the larger gain with adapting specialist behavior, while textual policies supply an additional correction channel as shown in Table 18.

In the full architecture, D-LRPO contributes 0.0251 and CB-SPO contributes 0.0098 relative to the frozen team. Joint adaptation increases the score to 0.4730, a gain of 0.0438 over the frozen team and 0.0453 over chief-only inference. The latter corresponds to a 10.6% relative improvement. Within

Table 18: **Architecture-by-adaptation configuration matrix on MedEHR-Common.** S, C, D, and B denote routed specialists, inference-time critique, D-LRPO, and CB-SPO. Δ is relative to the named reference. Output tokens count evaluation-time generation. D-LRPO and CB-SPO are undefined without active specialist state.

ID	Configuration	S	C	D	B	Reference	Sample-F1	Δ	95% interval	Output tokens/case
A0	Chief only	0	0	N/A	N/A	–	0.4277 ± 0.0038	–	–	2,040
A1	Chief + critic	0	1	N/A	N/A	A0	0.4408 ± 0.0040	$+0.0131$	$[+0.0040, +0.0222]$	2,510
A2	Chief + frozen specialists	1	0	0	0	A0	0.4260 ± 0.0049	-0.0017	$[-0.0135, +0.0101]$	2,950
A3	Frozen full team	1	1	0	0	A2	0.4292 ± 0.0049	$+0.0032$	$[-0.0046, +0.0110]$	3,620
A4	Specialists + D-LRPO	1	0	1	0	A2	0.4450 ± 0.0046	$+0.0190$	$[+0.0079, +0.0301]$	2,820
A5	Specialists + CB-SPO	1	0	0	1	A2	0.4300 ± 0.0048	$+0.0040$	$[-0.0051, +0.0131]$	2,780
A6	Specialists + both updates	1	0	1	1	A2	0.4546 ± 0.0047	$+0.0286$	$[+0.0162, +0.0410]$	2,720
A7	Full team + D-LRPO	1	1	1	0	A3	0.4543 ± 0.0045	$+0.0251$	$[+0.0140, +0.0362]$	3,440
A8	Full team + CB-SPO	1	1	0	1	A3	0.4390 ± 0.0047	$+0.0098$	$[+0.0007, +0.0189]$	3,360
A9	Full MedMDT	1	1	1	1	A3	$\mathbf{0.4730 \pm 0.0045}$	$+0.0438$	$[+0.0314, +0.0562]$	3,250

All performance entries summarize three runs of the corresponding configuration. Adaptation factors are evaluated for configurations with active specialists; A0 and A1 provide chief-only architectural controls. This is a constrained configuration matrix, not a complete four-factor design. Output tokens summarize evaluation-time generation per case.

Table 19: **Architecture–adaptation interaction contrasts.** Effects are contrasts of the three-run mean scores. Adjusted p-values apply Bonferroni correction to the three interaction tests. Intervals are pointwise.

Interaction	Effect	95% interval	p	p_{adj}
D-LRPO $\times$ CB-SPO	$+0.0089$	$[+0.0024, +0.0154]$	0.0076	0.0228
Critic $\times$ D-LRPO	$+0.0061$	$[-0.0004, +0.0126]$	0.0672	0.2017
Critic $\times$ CB-SPO	$+0.0058$	$[-0.0007, +0.0123]$	0.0819	0.2456

the three full-team adaptation-versus-frozen contrasts, Holm-adjusted p-values are 1.89×10^{-5} for D-LRPO, 0.0357 for CB-SPO, and 1.40×10^{-11} for joint adaptation.

Critique after specialist adaptation. Adding the critic changes the mean by 0.0032 when specialists are frozen and by 0.0184 when both adaptation channels are active. This pattern is consistent with critique becoming more useful when specialist proposals are better aligned with the task: verification can refine relevant candidate sets rather than compensating for poorly aligned proposals. The critic interactions with each individual update are evaluated separately below.

Complementarity and super-additivity. The additive reference for the two updates is

$$F_{\text{add}} = F_D + F_B - F_0 = 0.4543 + 0.4390 - 0.4292 = 0.4641. \tag{17}$$

The joint configuration exceeds this reference by

$$I_{D,B} = F_{D,B} - F_D - F_B + F_0 = 0.4730 - 0.4543 - 0.4390 + 0.4292 = 0.0089. \tag{18}$$

The interaction accounts for approximately 20.3% of the total joint improvement over the frozen team. CB-SPO contributes 0.0187 when D-LRPO is active, compared with 0.0098 without D-LRPO. Conversely, D-LRPO contributes 0.0340 when CB-SPO is active, compared with 0.0251 without CB-SPO. These conditional gains characterize complementary effects of the two adaptation channels.

For evaluation instance i in run r, the paired interaction contrast is

$$d_{i,r} = f_{i,r,D,B} - f_{i,r,D} - f_{i,r,B} + f_{i,r,0}. \tag{19}$$

Its variability includes the covariance among the four configurations and is distinct from the across-run standard deviation of an aggregate score.

The critic–D-LRPO contrast is $(A7 - A3) - (A4 - A2) = 0.0061$, with CB-SPO disabled. The critic–CB-SPO contrast is $(A8 - A3) - (A5 - A2) = 0.0058$, with D-LRPO disabled. The D-LRPO–CB-SPO interaction is evaluated with the critic enabled.

Table 20: **Inference resource use in the direct EHR-agent comparison.** Input and output tokens are evaluation-time averages per Common case. The methods share the chief backbone, temporally admissible EHR view, candidate vocabulary, and adaptation-data access.

System	Input tokens/case	Output tokens/case
ColaCare–MedEHR	∼27,000	∼3,650
EHRAgent–MedEHR	∼20,000	∼2,850
DynamiCare–MedEHR	∼29,000	∼3,900
MedMDT	∼26,000	∼3,250

Aggregate input and output caps are 131,072 and 8,192 tokens per case. These shared caps define a budget-capped comparison; realized token use differs across systems.

Table 21: **Direct comparison under the common MedEHR evaluation protocol.** Δ_{C} is the Common-set difference relative to MedMDT; negative values favor MedMDT. p_{adj} applies Holm correction over the three direct comparisons. Intervals are pointwise.

System	Common	Rare	Extreme	RCT	Δ_{C}	95% interval	p_{adj}
ColaCare–MedEHR	0.4368 ± 0.0065	0.3145 ± 0.0078	0.2814 ± 0.0069	0.5884 ± 0.0088	-0.0362	$[-0.0525, -0.0199]$	2.80×10^{-5}
EHRAgent–MedEHR	0.4186 ± 0.0072	0.2991 ± 0.0082	0.2708 ± 0.0075	0.5737 ± 0.0095	-0.0544	$[-0.0708, -0.0381]$	2.00×10^{-10}
DynamiCare–MedEHR	0.4462 ± 0.0063	0.3302 ± 0.0069	0.2893 ± 0.0062	0.6021 ± 0.0081	-0.0268	$[-0.0430, -0.0105]$	0.0013
MedMDT	$\mathbf{0.4730 \pm 0.0045}$	$\mathbf{0.3450 \pm 0.0061}$	$\mathbf{0.3004 \pm 0.0064}$	$\mathbf{0.6126 \pm 0.0072}$	0	–	–

The update interaction remains positive after correction in Table 19, whereas the two critic–update interactions remain unresolved. The distinction supports a focused interpretation: the strongest interaction evidence concerns the combination of parametric and textual adaptation. The positive interaction describes the combined gain in this evaluation; the nonsignificant critic interactions do not establish an absence of critic benefit. The 0.0089 interaction is calculated entirely from the three-run means in Table 18.

Inference resources. The full system uses 3,250 output tokens per case, compared 2,040 tokens for chief-only inference. Relative to the frozen full team, joint adaptation reduces output tokens by approximately 10.2%. This reduction accompanies the higher Sample-F1 of the jointly adapted team. Output tokens describe inference workload; they do not establish equal total compute across differently sized models.

E.2 Direct EHR-Agent Comparisons

Each reimplementation receives the same temporally admissible EHR information, candidate vocabulary, chief backbone, and evaluation interface. The same 600-case adaptation partition is available to every method. For methods without a native update mechanism, adaptation-partition access is limited to development choices within the comparison protocol.

Comparison across evaluation settings. DynamiCare is the strongest comparator in this comparison, reaching 0.4462 on Common. MedMDT improves on it by 0.0268, approximately 6.0% relative. The differences against ColaCare and EHRAgent are 0.0362 and 0.0544, respectively. All three Common-set comparisons retain significance after the specified correction.

The differences against DynamiCare decrease to 0.0148 on Rare, 0.0111 on Extreme, and 0.0105 on RCT. This pattern places the largest benefit in the development-aligned setting, while maintaining smaller gains under rare-label, dataset-shift, and trial-derived evaluation. The comparator ordering concerns the specified MedEHR reimplementations and does not constitute a ranking of the original systems on their native benchmarks as shown in Table 20 and Table 21.

Performance and resource use. MedMDT combines the highest Common score in the table with fewer output tokens than ColaCare and DynamiCare. EHRAgent uses fewer output tokens but has a lower score. This characterizes an accuracy–token trade-off within the specified resource envelope. The shared resource caps permit comparison within a common budget envelope; they do not equate realized computation or adaptation cost.

Table 22: **D-LRPO group-size and advantage-normalization sensitivity.** Δ is relative to the $G = 2$ standardized control in the same budget regime. Intervals are pointwise; p_{adj} uses Holm correction over the five non-control comparisons. GPU-hours count the single-B200 trainer allocation only.

Budget regime	Advantage rule	G	Attempts	Completions	Common F1	Δ	95% interval	p_{adj}	Accept rate	GPU-hours
Fixed updates	Standardized	2	168	1,344	0.4730 ± 0.0045	0	–	–	64%	8.4
Fixed updates	Standardized	4	168	2,688	0.4810 ± 0.0041	$+0.0080$	$[+0.0002, +0.0158]$	0.1820	70%	14.3
Fixed updates	Standardized	8	168	5,376	0.4842 ± 0.0044	$+0.0112$	$[+0.0021, +0.0203]$	0.0820	73%	25.2
Fixed rollouts	Standardized	2	168	1,344	0.4730 ± 0.0045	0	–	–	64%	8.4
Fixed rollouts	Standardized	4	84	1,344	0.4772 ± 0.0042	$+0.0042$	$[-0.0023, +0.0107]$	0.6230	68%	7.2
Fixed rollouts	Standardized	8	42	1,344	0.4741 ± 0.0048	$+0.0011$	$[-0.0067, +0.0089]$	1.0000	64%	6.4
Fixed rollouts	Centered, not scaled	2	168	1,344	0.4708 ± 0.0050	-0.0022	$[-0.0087, +0.0043]$	1.0000	61%	8.4

Trainer allocations exclude team inference and rule-proposal costs.

E.3 D-LRPO Group Size and Advantage Normalization

The optimizer analysis separates adapter-update attempts from sampled completions. Each attempt processes four buffered prompts with one generation pass per prompt. With U attempts and group size G, the completion count is

$$N_{\mathrm{completion}} = 4UG. \tag{20}$$

An attempt denotes a buffered adapter-training job rather than one optimizer step.

The fixed-update block uses 168 attempts, or 672 role-prompt uses. The fixed-rollout block holds the total at 1,344 completions as shown in Table 22.

Additional sampling at fixed update count. Increasing group size from two to four raises Sample-F1 by 0.0080. Increasing it further to eight adds 0.0032. The additional gain decreases from 0.0080 to 0.0032, while trainer allocation increases from 8.4 to 14.3 and then 25.2 GPU-hours. Adapter acceptance rises from 64% to 73%.

The unadjusted p-values are 0.0455 and 0.0164 for the $G = 4$ and $G = 8$ fixed-update comparisons. Neither passes the specified five-comparison correction. Thus, the numerical trend favors larger groups, while the precision is insufficient to establish every optimizer improvement as a separate corrected finding.

Equal-rollout comparison. With the completion budget fixed, $G = 4$ reaches 0.4772, whereas $G = 8$ reaches 0.4741. At a fixed completion budget, increasing group size reduces the number of update attempts, changing both within-prompt sampling and sequential adaptation frequency. The intervals include zero for both contrasts.

The $G = 4$ configuration offers the strongest numerical balance between score and trainer allocation in this block. Neither equal-rollout group-size comparison establishes a statistically significant improvement over $G = 2$.

Advantage normalization. Centering without scaling produces a difference of -0.0022. The interval includes zero, so this comparison does not resolve a performance difference. Removing normalization also changes the effective optimization scale; the contrast combines reward weighting and update-scale effects.

E.4 CB-SPO Exploration and Rule Coverage

The rule-coverage experiment uses a 60-rule comparison library, a 600-case adaptation stream, and 1,200 rule–case exposures. Exposure counts record cases in which a rule is active and receives reward credit. The suffix budget is 3,000 characters. Dynamic-rule promotion is assessed over 24 trials under the same acceptance gate as shown in Table 23.

Exploration coverage. Finite-priority initialization yields mean coverage of 80.0%. Both forced burn-in and infinite initialization reach 100.0% coverage. Forced burn-in gives each eligible rule one trial before reuse, while infinite initialization prioritizes unvisited rules. The comparison demonstrates that explicit exploration expands coverage under the same total exposure budget.

Table 23: **CB-SPO exploration and rule coverage.** Coverage and dynamic-rule promotion rates are averaged across runs. Each run uses 60 eligible rules, 600 adaptation cases, 1,200 rule–case exposures, and 24 dynamic-rule trials. Δ is relative to finite-priority initialization. Intervals are pointwise; p_{adj} uses Holm correction over the two comparisons.

Selection policy	Coverage (%)	Promotion (%)	Common F1	Δ [95% interval]	p_{adj}
Finite priority (0.25)	80.0	37.5	0.4730 ± 0.0045	0 [–]	–
Forced burn-in	100.0	45.8	0.4772 ± 0.0043	$+0.0042$ $[-0.0017, +0.0101]$	0.3230
Infinite initialization	100.0	41.7	0.4763 ± 0.0046	$+0.0033$ $[-0.0032, +0.0098]$	0.3230

Table 24: **RCT evaluation label distributions.** Positive patients have at least one action or adverse-event label. A patient can have multiple positive labels.

Task	Patients	Positive patients	No-action/no-event	Positive assignments
Care Actions 24h	500	382 (76.4%)	118 (23.6%)	1,094
Outcomes 14d	500	64 (12.8%)	436 (87.2%)	85
Residual Outcomes 24h	500	66 (13.2%)	434 (86.8%)	84

Performance and promotion. Forced burn-in increases Sample-F1 by 0.0042 and infinite initialization by 0.0033. The numerical gains accompany a higher dynamic-rule promotion rate, consistent with broader coverage supplying more diverse error-correction opportunities. The corrected p-values remain above 0.05.

The main observed distinction is comprehensive coverage with small numerical increases in accuracy. These values do not establish that every additional explored rule is useful, or that static-rule coverage directly causes dynamic-rule acceptance.

E.5 RCT Label Distribution and Positive-Class Evaluation

Evaluation cohorts. Each RCT task contains 500 instances from 500 distinct subjects. Across the three tasks, there are 862 distinct subjects. Pairwise subject overlaps are 248 between Care Actions and Outcomes, 274 between Care Actions and Residual Outcomes, and 243 between Outcomes and Residual Outcomes. These are counts of evaluation subjects, not independent training repetitions. Each task is evaluated separately below.

Outcomes contains 49 AKI progression, 11 dialysis, and 25 death labels. Residual Outcomes contains 45 AKI progression, nine dialysis, and 30 death labels. Their sums exceed the number of positive patients because events overlap. Care Actions contains 12 positive candidates, with frequencies ranging from 321 patients for urine-output measurement to seven for contrast-related procedures and one for aminoglycoside administration.

Metric scope. Explicit no-event labels are frequent in both outcome tasks. Consequently, overall Sample-F1 is strongly influenced by performance on no-event cases and cannot alone establish adverse-event detection ability. Positive-class metrics evaluate the action and adverse-event labels separately from these negative labels. They complement the primary metric and do not establish superiority over other systems without corresponding positive-class comparisons.

For the default MedMDT configuration, positive-class macro precision, recall, and F1 average over the 12 action candidates or three adverse-event candidates, excluding the explicit no-action or no-event label. Metrics are computed within each run and then averaged across the three runs. Macro F1 averages candidate-specific F1 values; it is not the harmonic mean of the reported macro precision and recall.

Macro AUPRC averages candidate-specific precision–recall areas. Brier score averages squared probability errors over patients and positive candidates. ECE_{15} averages candidate-specific calibration errors over 15 equal-width probability bins. These metrics evaluate candidate-level probability scores, whereas Sample-F1 evaluates the final predicted label sets.

Table 25: **Positive-class, calibration, and selective-prediction results for default MedMDT on MedEHR-RCT.** Entries are three-run means. AUPRC, Brier, and ECE are computed over positive candidates. Higher precision, recall, F1, and AUPRC are better; lower Brier and ECE are better. Coverage uses the task-specific risk targets in the text.

Task	Macro precision	Macro recall	Macro F1	AUPRC	Brier	ECE_{15}	Risk target	Coverage
Care Actions 24h	0.317	0.395	0.335	0.395	0.103	0.042	0.60	0.460
Outcomes 14d	0.328	0.585	0.390	0.390	0.041	0.032	0.15	0.680
Residual Outcomes 24h	0.414	0.595	0.465	0.435	0.038	0.027	0.10	0.760

Selective risk is the retained-case mean $1 - \text{Sample-F1}$. The task-specific risk targets are 0.60 for Care Actions, 0.15 for Outcomes, and 0.10 for Residual Outcomes. Coverage is the retained fraction at the corresponding target and is averaged across runs.

Action and adverse-event prediction. Care Actions has macro F1 of 0.335 and macro AUPRC of 0.395. Recall exceeds precision, indicating that action recovery is accompanied by false-positive predictions. Candidate imbalance makes the macro metrics sensitive to errors on rare actions, particularly the action observed in only one evaluation patient.

Outcomes has macro recall of 0.585, precision of 0.328, and F1 of 0.390. These values demonstrate recovery of positive adverse-event labels while also identifying a substantial false-positive burden. Residual Outcomes reaches precision of 0.414, recall of 0.595, and F1 of 0.465. Its inputs include first-day information available at the 24-hour landmark; the predicted events occur after that landmark. The two outcome tasks differ in sampled subjects and eligibility, so their score difference does not isolate the contribution of the additional 24-hour history.

Probability scores and selective prediction. Brier scores are 0.103, 0.041, and 0.038, and ECE values are 0.042, 0.032, and 0.027 for Care Actions, Outcomes, and Residual Outcomes, respectively. These summarize probabilistic errors within each task; differences across tasks also reflect prevalence and candidate composition. Mean coverage is 46.0%, 68.0%, and 76.0% at the respective risk targets. Since both the risk targets and case distributions differ, coverage is not a common-threshold ranking of the tasks. In the outcome tasks, retained-case accuracy can be dominated by no-event cases; coverage alone does not establish preservation of positive-event sensitivity.

E.6 Specialist Disagreement and Chief Resolution

For each Procedures case, disagreement is the Jaccard distance between the two specialist proposal sets. In run r, let $f_{i,r}^{\text{final}}$ be final Sample-F1 and $f_{i,r}^{a}$ and $f_{i,r}^{b}$ the specialist scores. Define

$$\text{DRG}_{i,r} = f_{i,r}^{\text{final}} - \max\{f_{i,r}^{a}, f_{i,r}^{b}\}. \tag{21}$$

Strict resolution success corresponds to $\text{DRG}_{i,r} > 0$, and non-degradation to $\text{DRG}_{i,r} \geq 0$.

Aggregation across runs. Disagreement and bin membership are determined from the proposals in each run. Each run contributes its within-bin mean F1, mean DRG, resolution success rate, and non-degradation rate; Table 26 reports their arithmetic averages over the three runs. Thus, a displayed percentage is a mean of run-specific proportions, not a success count divided by a single fixed bin size. The 200 Procedures cases in each subset form the evaluation cohort; their bin memberships can vary across runs. The table reports descriptive means without count-based binomial intervals or an aggregate reconstructed from a single bin allocation.

Common. Mean DRG reaches 0.0490 in the $[0.2, 0.4)$ bin, which also has the highest final F1 (0.6094) and RSR (65.6%). In the highest-disagreement bin, final F1 is 0.5726 and mean DRG is 0.0210. These descriptive means indicate that more disagreement does not necessarily produce larger reconciliation gains.

Rare. The largest mean DRG, 0.0380, occurs at disagreement $[0.4, 0.6)$, with final F1 of 0.4842 and RSR of 59.8%. The peak occurs at a higher disagreement range than on Common. Mean DRG remains positive in the highest bin but falls to 0.0150. This pattern is consistent with useful proposal diversity at intermediate disagreement and less effective reconciliation at the highest disagreement.

Table 26: **Three-run mean disagreement and resolution results on Procedures.** Best-specialist F1 is the within-case maximum of the two specialist scores, averaged within each bin and then across runs. Final F1 and mean DRG follow the same aggregation. RSR and non-degradation are mean within-run percentages. Differences across bins are descriptive.

Subset	Disagreement bin	Best-specialist F1	Final F1	Mean DRG	RSR (%)	Non-degradation (%)
Common	$[0, .2)$	0.5601	0.5971	0.0370	61.8	89.0
Common	$[.2, .4)$	0.5604	0.6094	0.0490	65.6	92.0
Common	$[.4, .6)$	0.5497	0.5957	0.0460	64.1	90.0
Common	$[.6, .8)$	0.5522	0.5892	0.0370	62.3	87.0
Common	$[.8, 1]$	0.5516	0.5726	0.0210	59.4	80.0
Rare	$[0, .2)$	0.4349	0.4549	0.0200	52.7	86.0
Rare	$[.2, .4)$	0.4425	0.4735	0.0310	56.9	89.0
Rare	$[.4, .6)$	0.4462	0.4842	0.0380	59.8	91.0
Rare	$[.6, .8)$	0.4387	0.4697	0.0310	57.4	87.0
Rare	$[.8, 1]$	0.4270	0.4420	0.0150	51.8	78.0
Extreme	$[0, .2)$	0.1183	0.1243	0.0060	32.6	84.0
Extreme	$[.2, .4)$	0.1194	0.1314	0.0120	35.4	86.0
Extreme	$[.4, .6)$	0.1242	0.1422	0.0180	38.7	88.0
Extreme	$[.6, .8)$	0.1241	0.1371	0.0130	36.9	83.0
Extreme	$[.8, 1]$	0.1085	0.1125	0.0040	31.3	75.0

Extreme. Mean DRG is positive in every bin but ranges only from 0.0040 to 0.0180. At moderate disagreement, RSR is 38.7% and non-degradation is 88.0%. Preservation is therefore more frequent than strict improvement, while the low final F1 indicates substantial remaining difficulty under dataset shift.

Interpretation. Positive DRG measures improvement over the stronger specialist. Non-degradation also counts preservation of that specialist's score. Their separation distinguishes improvement from maintenance of an existing answer. The binwise patterns describe reconciliation behavior. They do not identify an optimal disagreement level or establish that inducing disagreement improves performance.

F Discussion

F.1 Interpretation of the Evidence

The most informative comparisons are those that hold the chief backbone and evaluation protocol fixed. Under that control, the gap between the full MedMDT system, MedMDT mini, and the listed same-chief agentic baselines isolates the value of the structured team more directly than comparisons among different frontier model families. The component ablations further show that no single mechanism explains the full endpoint: removing specialist adaptation, semantic-policy optimization, or critique reduces the reported average, while the disagreement and transition analyses show that correction involves both false-positive removal and false-negative recovery. These observations support complementarity, not a claim that every component is necessary for every task or patient.

The checkpoint curves are also consistent with useful between-case adaptation: performance rises as the number of adaptation cases increases on both Common and RCT. They should not be read as a theorem of monotonic online improvement. A checkpoint aggregates a changing sequence of adapters, skills, and cases. Likewise, superior performance on the RCT-derived tasks does not imply that the system estimates a causal treatment effect. The tasks predict recorded care actions and outcomes under the trial's observed data-generating process.

The dual-state design offers a practical division of labor. Role-local adapters can absorb task-specific response patterns that are difficult to express as short rules, while semantic skills are inspectable, inexpensive to roll back, and portable across compatible model checkpoints. Their combination is most useful when the two channels remain identifiable. MedMDT therefore freezes competing updates during a dynamic-rule trial, versions accepted state, and separates adaptation from evaluation. These controls reduce–but do not eliminate–nonstationarity and credit-assignment ambiguity.

F.2 Failure Modes and Limitations

Correlated reasoning errors. Specialist labels do not guarantee diverse evidence or independent errors because roles can share a pretrained backbone, prompt conventions, and retrieved context. A persuasive but incorrect proposal can propagate through the critic and chief. Future evaluations should report proposal diversity, error correlation, and performance under deliberately corrupted specialists in addition to aggregate F1.

Imperfect credit assignment. D-LRPO trains role-local completions using shaped rewards, whereas CB-SPO assigns the shared case reward to all active rules. Neither mechanism identifies an individual agent's causal contribution. Small GRPO groups can yield weak relative advantages, and sequential eight-case rule windows can accept changes caused by case-mix variation. Larger paired or randomized validation windows would provide stronger evidence, at greater data and compute cost.

Dataset dependence. MIMIC represents critical-care practice at particular institutions and eras, and ELAIA is a multicenter AKI-alert trial with its own eligibility and documentation process. The Extreme subset probes one form of schema and dataset shift, not geographic, demographic, or prospective generalization.

Metric and calibration limits. Sample-F1 weights cases equally and can reward plausible set overlap without measuring ranking, probability calibration, clinical severity, or downstream utility. The explicit negative labels in RCT tasks and the dual raw/canonical representation for laboratory labels also affect task difficulty. Confidence fields are model-generated text, not calibrated probabilities. No claim is made about decision-curve benefit, safety-critical recall, subgroup parity, or clinician time saved.

Model and system reproducibility. Closed or rapidly changing model endpoints can drift even under the same product name. Exact model revision identifiers, decoding parameters, prompts, accepted adapter hashes, skill-registry versions, and per-case outputs are necessary for archival reproducibility; Appendix G specifies the corresponding versioned state manifest.

F.3 Clinical Scope

MedMDT is a retrospective research system. It does not recommend treatment, replace a multidisciplinary team, or establish effectiveness in a live workflow. Before any prospective evaluation, a locked version would require independent external validation, predefined failure handling, calibrated abstention, subgroup and site analyses, privacy and security review, human-factors testing, and continuous monitoring. Reporting guidance for clinical LLM studies and trustworthy medical AI emphasizes the same separation between technical benchmark performance and deployment evidence (Gallifant et al., 2025; Wiens et al., 2019; Lekadir et al., 2025). Early live evaluation would require a prospectively specified protocol and stage-appropriate reporting such as DECIDE-AI (Vasey et al., 2022).

G Operational Protocol

G.1 Case Lifecycle

The following procedure makes the inference/adaptation boundary explicit. "Adapt" and "evaluate" are mutually exclusive run modes for a case stream.

1. **Load versioned state.** Resolve the chief and critic revisions, each role's adapter hash, the skill-registry version, candidate-vocabulary hash, prompt hash, and runtime configuration. In evaluation mode, reject any configuration that enables training or prompt mutation.
2. **Construct the admissible view.** Read the task, prediction time, candidate vocabulary, and only the observations permitted before the task horizon. Remove the target and fields on the deny list before serialization.

3. **Retrieve and summarize.** The retriever executes supported time-filtered queries. The chief forms a private summary that is not copied into the specialist prompts verbatim.
4. **Route specialists.** Activate the fixed task-relevant roles from Table 11. Each specialist receives the same admissible case boundary but a role-specific prompt, adapter, and semantic suffix.
5. **Critique and commit.** The critic compares proposals with admissible evidence and returns supported, unsupported, corrected, and missing candidates. The chief emits one normalized candidate set. Persist the final-decision event before requesting or reading the target.
6. **Score.** Transform prediction and target with the task-aware evaluator and compute case Sample-F1. Store both raw and normalized sets so that scoring can be audited.
7. **Adapt or stop.** In evaluation mode, stop without mutation. In adaptation mode, release the scored event to eligible role buffers and CB-SPO. If a dynamic-rule trial is active, hold competing updates fixed; otherwise schedule configured adapter fitting and rule reselection.
8. **Validate and register.** Validate a candidate state, atomically register accepted versions, and preserve the parent version and reason for acceptance or rollback. The next case can read only a completely registered state.

G.2 Protocol Invariants

Each evaluation or adaptation checkpoint should be reconstructible from one manifest containing the fields below. Hashes refer to immutable bytes rather than filenames as shown in Table 27 and Table 28.

Table 27: Operational invariants required by the claimed evaluation protocol.

Invariant	Required behavior
Temporal boundary	Every retrieved row and derived feature obeys the task-specific horizon; unknown or unsafe times are excluded.
Label privacy	Gold targets and target-derived summaries are absent from every pre-commit message and tool response.
Prediction commitment	The normalized final candidate set is persisted before scoring or feedback release.
Role ownership	A D-LRPO update can write only the designated role's adapter registry entry.
Trial isolation	While a CB-SPO candidate is tested, relevant adapter updates and prompt reselection are deferred.
Frozen evaluation	Model revisions, adapters, skills, prompts, candidates, and decoding settings remain fixed for the complete evaluation stream.
Atomic versioning	A state becomes readable only after its manifest, files, and hashes are complete; rollback restores the recorded parent.
Auditability	Raw/normalized predictions, tool queries, state identifiers, and scoring outputs share a case and correlation identifier.

Table 28: Minimum state manifest for a reproducible MedMDT run.

Group	Required fields
Run identity	Run ID, parent run, mode, start/end time, code commit, environment lock hash, command-line arguments
Data	Subset and split-manifest hashes, patient/case counts, task counts, candidate-vocabulary hash, feature-whitelist hash
Models	Provider or repository, exact revision, tokenizer revision, quantization, context limit, decoding parameters
Team state	Chief/critic prompt hashes; for every role, base revision, adapter hash, parent adapter, skill-registry version and suffix hash
Optimization	Buffer case IDs, group size, reward-function version, optimizer settings, seeds, acceptance score, rollback reason
Evaluation	Frozen-state assertion, raw and normalized predictions, targets stored separately, per-case score, failure/timeout status
Resources	Hardware allocation, wall-clock time, input/output tokens, peak memory, energy or API cost when available

G.3 Prompt and Output Contracts

The following are schema-level templates, included to clarify information flow without claiming that wording alone guarantees compliance. An archival artifact should contain the verbatim prompts and their hashes.

Specialist contract. The prompt specifies: (i) role and task; (ii) prediction time and admissible evidence; (iii) the complete task-level candidate vocabulary; (iv) a versioned, bounded role/task skill suffix; and (v) a structured response with `predictions`, `rationale`, and `confidence`. It instructs the model to return only candidate labels, cite time-valid evidence in the rationale, avoid unsupported additions, and express uncertainty without inventing a new label.

Critic contract. The critic receives the task, prediction time, admissible evidence, candidate vocabulary, and specialist proposals. Its response contains four disjoint review lists–`supported`, `unsupported`, `corrected`, and `missing_supported`–plus concise concerns. The critic cannot see the gold target. Corrections must map back to the candidate vocabulary and identify supporting evidence.

Chief contract. The chief receives its private summary, specialist proposals, and critique. It must produce one normalized `final_predictions` list and a concise synthesis, resolve duplicates and aliases, and avoid labels outside the candidate vocabulary. The chief is instructed that critique is advisory rather than authoritative and that unsupported consensus remains unsupported.

Dynamic-skill contract. The text optimizer receives aggregate task/role error descriptors and summary metrics, not patient records or target strings. It proposes at most two generic rules within the configured character limit. Rules must be prospective, task-relevant, free of patient-specific content, and executable using only information available at inference. Examples include mapping medication formulations to generic drug labels (therapy agent) and prioritizing time-valid direct evidence when resolving disagreements (chief). These concise, role-specific instructions make the guidance available for adaptation human-readable and directly inspectable. The system rejects rules that request labels, future events, identifiers, hidden chain-of-thought, or external data not permitted by the runtime.