LONGRX: CLOSED-LOOP MEDICATION RECOMMENDATION WITH ACTION-COVERAGE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Medication recommendation is commonly formulated as one-shot set prediction, although prescription construction naturally involves iterative information acquisition, medication adjustment, and drug–drug interaction verification. We propose **LongRx**, a closed-loop medication decision agent that sequentially constructs prescriptions through five parameterized actions: `RETRIEVE`, `ADD`, `DELETE`, `VERIFY`, and `EXIT`. Starting from an empty candidate set and prescription, LongRx learns state-dependent decision sequences rather than following a predefined workflow. To improve sequential learning, we combine terminal prescription supervision with process-level feedback on local prescription changes. Our **Coverage-Aware Tree Optimization (CATO)** combines opportunity-aware action coverage with tree-relative credit, broadening exploration of under-explored valid decisions and comparing their downstream outcomes from the same prescription state, while concrete medication edits and retrieval queries remain policy-generated. In an idealized fixed-state, single-draw model, the coverage correction contracts pairwise action log-odds. Experiments on MIMIC-III, MIMIC-IV, and eICU show improved overlap with recorded prescriptions and lower rates of DrugBank-listed interacting medication pairs relative to RES-MR. These metrics characterize agreement with historical prescribing and a database-based interaction proxy; they do not establish clinical appropriateness or improved patient outcomes.Our code is available at https://anonymous.4open.science/r/LongRx.

1 INTRODUCTION

Medication recommendation aims to predict an appropriate medication set from patient EHRs while controlling drug–drug interaction (DDI) risk Ali et al. (2023); Zhang et al. (2026b); Stark et al. (2019). Existing methods have incorporated longitudinal histories, DDI information, molecular structures, and, more recently, LLM-based clinical representations Shang et al. (2019); Yang et al. (2021b); Wu et al. (2022); Yang et al. (2021a; 2023); Zhao et al. (2025); Fan et al. (2026). Despite enabling iterative refinement, recent LLM-based approaches still typically organize medication decisions through predefined stages Fan et al. (2026); Xu & Li (2025); Zhao et al. (2025).

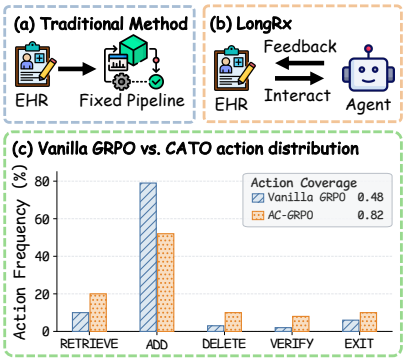

Figure 1: Motivation of LongRx. **(a–b)** Predefined workflow versus closed-loop decision making. **(c)** CATO mitigates action exploration collapse in Vanilla GRPO.

As illustrated in Figure 1(a)–(b), such workflows largely specify in advance when different operations should be performed. However, prescription construction is inherently state dependent: after each retrieval, medication edit, or verification result, the appropriate next decision may change. We therefore formulate medication recommendation as a closed-loop process in which the agent dynamically decides whether to retrieve candidates, add or remove medications, verify DDI risk, or terminate according to the evolving prescription state.

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.

Learning such a policy introduces a fine-grained credit-assignment problem. A terminal prescription reward evaluates the final outcome, but provides only coarse supervision for determining which intermediate decisions were responsible for that outcome. Moreover, comparing independently sampled trajectories confounds the current decision with differences in their preceding histories. A natural remedy is to expand multiple continuations from the same intermediate state, so that alternative decisions can be compared under an identical patient and prescription context.

However, tree-structured rollout alone does not guarantee diverse decision exploration. If sibling branches are still generated primarily from the current policy, high-probability actions can repeatedly dominate the tree, while valid but less frequent decisions such as `DELETE`, `VERIFY`, or repeated `RETRIEVE` remain weakly explored. As shown in Figure 1(c), this leads to *action exploration collapse*: under-explored actions receive little downstream supervision and consequently remain unlikely under subsequent policies. Thus, effective tree-based optimization requires not only comparing alternative decisions, but also ensuring that sufficiently diverse valid actions enter the comparison.

To address these challenges, we propose **LongRx**, a closed-loop medication decision agent that sequentially constructs prescriptions through five parameterized actions: `RETRIEVE`, `ADD`, `DELETE`, `VERIFY`, and `EXIT`. Starting from an empty candidate set and prescription, LongRx learns state-dependent decision sequences with a single policy. We provide process-level feedback for intermediate prescription edits and introduce **Coverage-Aware Tree Optimization (CATO)**, which expands alternative decisions from shared prescription states, promotes valid but under-explored action types through opportunity-aware coverage, and assigns tree-relative credit according to downstream outcomes. Across MIMIC-III, MIMIC-IV, and eICU, LongRx improves agreement with recorded prescriptions and lowers DrugBank-based DDI rates, while CATO yields broader action exploration than standard GRPO.

The main contributions are summarized as follows:

- We introduce **LongRx**, a closed-loop prescription agent that moves beyond predefined medication recommendation workflows by dynamically selecting retrieval, prescription editing, DDI verification, and termination according to the evolving state.
- We instantiate multi-level credit assignment for sequential prescription construction by combining terminal supervision, medication-set process feedback, and the shared-state sibling comparison used in tree-based GRPO.
- We propose **Coverage-Aware Tree Optimization (CATO)**, whose new component is opportunity-normalized branching over state-valid actions. It is coupled with established shared-state tree credit to test whether targeted coverage improves over temperature and entropy exploration under matched budgets.

2 Related Work

2.1 Medication Recommendation

Medication recommendation has progressed from visit-level prediction to longitudinal, knowledge-enhanced, and LLM-based modeling Ali et al. (2023); Zhang et al. (2026b); Zhu et al. (2024). Representative methods model label dependencies Zhang et al. (2017), longitudinal medication histories Shang et al. (2019); Wu et al. (2022); Yang et al. (2021a), molecular structures Yang et al. (2021b; 2023), and rare or multimodal patient representations Zhao et al. (2024); Tan et al. (2024). Recent LLM-based approaches explore knowledge distillation, clinical-text modeling, list-wise generation, and safety-grounded agents Liu et al. (2024); Zhao et al. (2025); Fan et al. (2026); Wang et al. (2026b). Unlike these methods, LongRx formulates medication recommendation as bounded interaction between a single agent and a patient-conditioned prescription environment, jointly learning set editing, safety verification, and timely termination.

2.2 LLM Agents and Long-Horizon Policy Optimization

LLM agents address complex tasks through iterative reasoning, tool use, and environment feedback, but sparse terminal rewards make long-horizon credit assignment difficult Du et al. (2026); Zhang

et al. (2025); Chang et al. (2026). Early studies establish reasoning–action interaction, browser navigation, and autonomous tool invocation Yao et al. (2022); Nakano et al. (2021); Schick et al. (2023). Agent-R1, Search-R1, and ReTool learn interactive policies from outcome-level feedback Cheng et al. (2025); Jin et al. (2025); Feng et al. (2026), whereas iStar, IGPO, HiPRAG, and WebArbiter provide finer supervision for intermediate decisions Liu et al. (2026); Wang et al. (2025); Wu et al. (2026); Zhang et al. (2026a). Tree-GRPO compare alternative actions from shared states, while ATPO, DAPO improve exploration or optimization stability Ji et al. (2026); Cao et al. (2026); Yu et al. (2026). LongRx adapts these advances to cost-effective medication-set editing, safety verification, and termination in a patient-conditioned prescription environment.

3 Problem Formulation

Let $\mathcal{M}$ be the medication vocabulary and $\mathcal{D} = \{(X, Y)\}$ a set of encounters, where X is the patient information available before the medication decision and $Y \subseteq \mathcal{M}$ is the recorded prescription. Rather than predicting a medication set from X in one shot, we let a policy build it step by step. At step t the agent observes

$$s_t = (X,\, M_t,\, C_t,\, V_t,\, H_t), \qquad M_0 = C_0 = V_0 = H_0 = \varnothing, \tag{1}$$

where M_t is the current prescription, C_t the candidates exposed so far, V_t the latest verification result still valid for M_t, and H_t the interaction history, and emits an action $u_t = (a_t, \xi_t)$ whose type a_t is one of `RETRIEVE`, `ADD`, `DELETE`, `VERIFY`, and `EXIT` (the set $\mathcal{A}$) and whose argument is ξ_t. `RETRIEVE` sends a generated query to a frozen retrieval backend and adds up to $K_{\text{ret}} = 16$ unseen medications to C_t; `ADD` moves a subset of $C_t \setminus M_t$ into M_t and `DELETE` removes a subset of M_t, either edit clearing V_t; `VERIFY` stores the DDI pairs of M_t found by a fixed checker; `EXIT` requests termination, which is accepted only after the deterministic terminal verification gate described in Appendix J; otherwise its feedback is returned to the policy for another decision. The accepted set is returned as M_T, and episodes are cut off after $T_{\max} = 8$ steps, at which point the same checker is run once before the current set is returned. Only the types in $\mathcal{A}_{\text{valid}}(s_t) \subseteq \mathcal{A}$ are executable, invalid outputs leave the state unchanged, and each action carries a cost $c(u_t)$ (one per tool call or invalid output, one per edited medication, zero for `EXIT`). Y never enters s_t and is used only for rewards and evaluation.

The policy π_θ is an LLM that emits the action as a JSON object and factorizes as $\pi_\theta(u_t \mid s_t) = \pi_\theta(a_t \mid s_t)\, \pi_\theta(\xi_t \mid s_t, a_t)$. We optimize it with GRPO Shao et al. (2024): for each encounter, G trajectories are sampled from $\pi_{\theta_{\text{old}}}$, every step of trajectory i receives the group-standardized terminal advantage $A_i^{\text{out}} = (R_i - \text{mean}_j\, R_j)/(\text{std}_j\, R_j + \epsilon)$, and the policy is updated with the clipped PPO surrogate plus a KL penalty toward a reference model. CATO (Section 4.2) keeps this objective and changes how the G trajectories are generated and how the per-step advantage is formed.

4 Method

In this section, we present LongRx; Figure 2 gives an overview. Section 4.1 defines the terminal and process-level credit used to train the policy of Section 3, and Section 4.2 presents *Coverage-Aware Tree Optimization* (CATO), which generates rollout trees with coverage-aware branching and assigns tree-relative credit. Before reinforcement learning, the policy is warm-started on trajectories generated by a stronger teacher with the same prompt and tools and without access to Y: for each training encounter, the teacher samples eight trajectories, the top quartile by terminal reward is kept, and the policy is fine-tuned on their action tokens (Appendix F).

4.1 Multi-Level Prescription Credit

Training signals for LongRx come from the recorded prescription Y, which is compared against the agent's output only after an episode ends. A single terminal score is the natural objective, but an episode consists of several edits, retrievals, and checks, and one number cannot tell which of them helped. We therefore provide credit at two granularities: a terminal reward on the completed prescription and a process signal on each individual edit. For a completed trajectory, the terminal

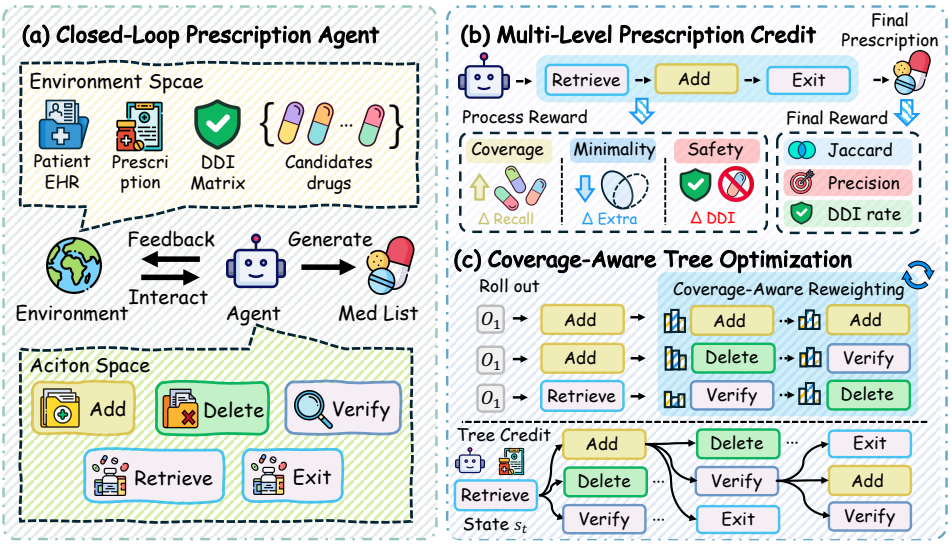

Figure 2: Overview of LONGRX. **(a)** The agent constructs a prescription through a closed loop of high-level actions, each instantiated with a policy-generated argument. **(b)** Multi-level credit combines terminal prescription supervision with localized process feedback for medication edits. **(c)** CATO promotes under-explored valid actions through opportunity-aware branching and credits each decision by comparing the downstream outcomes of sibling branches expanded from the same state.

reward is

$$R^{\text{final}} = w_f \ \text{F1}(M_T, Y) - w_d \ \text{DDI}(M_T) - w_m \, E_{\text{med}}(M_T, Y), \tag{2}$$

where $E_{\text{med}}(M, Y) = \big||M| - |Y|\big| / \max(1, |Y|)$ is the normalized prescription-size deviation and $(w_f, w_d, w_m) = (1.0, 0.5, 0.1)$, with F1 and DDI on the $[0, 1]$ scale.

The process signal is defined for prescription-modifying actions:

$$r_t^{\text{proc}} = \lambda_{\text{rec}} \Delta\text{Recall}_t + \lambda_{\text{red}} \Delta\text{Extra}_t + \lambda_{\text{safe}} \Delta\text{DDI}_t, \tag{3}$$

where, for an edit from M_t to M_{t+1}, the three terms are $\text{Recall}(M_{t+1}, Y) - \text{Recall}(M_t, Y)$, $(|M_t \backslash Y| - |M_{t+1} \backslash Y|) / \max(1, |Y|)$, and $\text{DDI}(M_t) - \text{DDI}(M_{t+1})$: the gain in reference-medication recall, the reduction of medications absent from the reference, and the reduction in DDI risk. We use $(\lambda_{\text{rec}}, \lambda_{\text{red}}, \lambda_{\text{safe}}) = (0.5, 0.25, 0.25)$ and clip the signal to $\hat{r}_t^{\text{proc}} = \text{clip}(r_t^{\text{proc}}, -1, 1)$. The process signal applies to `ADD` and `DELETE`, which change the prescription; `RETRIEVE` and `VERIFY` acquire information rather than editing the set and are credited through their downstream outcomes. The action cost $c(u_t)$ of Section 3 enters as a per-step penalty (Eq. (10)).

4.2 COVERAGE-AWARE TREE OPTIMIZATION

GRPO samples multiple trajectories independently from the current policy, but trajectory diversity does not necessarily imply decision diversity. In LongRx, high-probability patterns such as `RETRIEVE`→`ADD`→`EXIT` can dominate a rollout group, leaving valid but less frequent decisions such as `DELETE`, `VERIFY`, or repeated `RETRIEVE` under-explored. Tree expansion enables comparison between alternative decisions from the same prescription state, but branches sampled from the original policy can inherit the same bias. CATO therefore combines *coverage-aware branching*, which broadens the valid action types explored from a state, with *tree-relative credit*, which determines which explored decisions should be reinforced.

Rollout tree. For each encounter, CATO grows a tree rooted at s_0 (Algorithm 1), where nodes are states and edges are parameterized actions $u_t = (a_t, \xi_t)$. At each expanded state, up to B sibling actions are sampled and independently continued until `EXIT` or $T_{\max}$. The frontier is a first-in–first-out queue, so states are expanded in breadth-first order. To strictly enforce a terminal-trajectory

Algorithm 1 CATO rollout tree for one encounter

Require: $\pi_{\theta_{\text{old}}}$, branching factor B, leaf budget G, coverage strength β
$N_g(a), O_g(a) \leftarrow 0$ for all $a \in \mathcal{A}$; $\mathcal{F} \leftarrow \text{FIFO}(s_0)$; $\mathcal{L} \leftarrow \varnothing$
while $|\mathcal{L}| < G$ **do**
  if $\mathcal{F} = \varnothing$ **then**
    push a fresh copy of s_0 to $\mathcal{F}$
  end if
  pop the shallowest state s_t from the head of $\mathcal{F}$; $k_t \leftarrow \min\{B, G - |\mathcal{L}| - |\mathcal{F}|\}$
  for $j = 1, \ldots, k_t$ **do**
    increment $O_g(a)$ for all $a \in \mathcal{A}_{\text{valid}}(s_t)$
    draw $a_t \sim q_g^{(j)}(\cdot \mid s_t)$; increment $N_g(a_t)$
    draw $\xi_t \sim \pi_{\theta_{\text{old}}}(\cdot \mid s_t, a_t)$ and execute $u_t = (a_t, \xi_t)$
    add terminal successors to $\mathcal{L}$; otherwise push them to $\mathcal{F}$
  end for
end while
return $\mathcal{L}$ and sibling sets

budget G, we maintain

$$|\mathcal{L}| + |\mathcal{F}| \leq G, \tag{4}$$

where $\mathcal{L}$ and $\mathcal{F}$ are the completed leaves and active frontier. After popping a state, we expand at most

$$k_t = \min\{B,\ G - |\mathcal{L}| - |\mathcal{F}|\}, \tag{5}$$

so every active branch reserves one future leaf slot. If all active branches terminate before the budget is filled, a fresh copy of s_0 is inserted into the queue. Thus every variant, including $B = 1$, returns exactly G terminal trajectories and is compared under the same leaf budget. We use $B = 3$ and $G = 8$.

Coverage-aware branching. Different action types have different opportunities to be selected. For tree g, let $O_g(a)$ denote the number of sibling draws where a is valid and $N_g(a)$ the number of times it is selected. We define its empirical selection coverage as

$$c_g(a) = \frac{N_g(a) + \alpha}{O_g(a) + \alpha}. \tag{6}$$

A small $c_g(a)$ indicates an action that has often been available but remains under-explored. Sibling types are sampled from

$$q_g^{(j)}(a \mid s_t) \propto \mathbf{1}[a \in \mathcal{A}_{\text{valid}}(s_t)]\pi_{\theta_{\text{old}}}(a \mid s_t)c_g(a)^{-\beta}, \tag{7}$$

where α smooths the counts and β controls coverage correction. The original policy probability remains in the distribution, so CATO neither enforces uniform action usage nor fixed quotas. Coverage acts only on the high-level action type; its structured argument is still generated from $\pi_{\theta_{\text{old}}}(\xi_t \mid s_t, a_t)$ and never enters the reward.

Tree-relative credit. For branch u_t taken at s_t, let $\mathcal{L}(s_t, u_t)$ denote its descendant terminal leaves. Its downstream value is

$$Q_{\text{tree}}(s_t, u_t) = \frac{1}{|\mathcal{L}(s_t, u_t)|} \sum_{\ell \in \mathcal{L}(s_t, u_t)} R_\ell^{\text{final}}. \tag{8}$$

Branches from the same state are normalized against their siblings:

$$A_{\text{tree}}(s_t, u_t) = \frac{Q_{\text{tree}}(s_t, u_t) - \mu_{\text{sib}}(s_t)}{\sigma_{\text{sib}}(s_t) + \epsilon}. \tag{9}$$

We set $A_{\text{tree}} = 0$ for states with only one realized child. Because siblings share the same preceding state, this provides localized credit for the current decision. It is a lower-confounding comparison than independent trajectories, not a causal estimate: downstream stochasticity and the tree-induced state distribution remain part of the estimator.

Objective. For step t of trajectory i, we use

$$A_{i,t} = A_i^{\text{out}} + \lambda_{\text{tree}} A_{\text{tree}}(s_{i,t}, u_{i,t}) + \lambda_p \widehat{r}_{i,t}^{\text{proc}} - \lambda_c c(u_{i,t}). \tag{10}$$

Coverage-aware sampling changes the behavior distribution, so we separate its importance correction from the PPO policy ratio:

$$w_{i,t}^{\text{cov}} = \frac{\pi_{\theta_{\text{old}}}(a_{i,t} \mid s_{i,t})}{q_g^{(j)}(a_{i,t} \mid s_{i,t})}, \qquad r_{i,t}(\theta) = \frac{\pi_\theta(u_{i,t} \mid s_{i,t})}{\pi_{\theta_{\text{old}}}(u_{i,t} \mid s_{i,t})}. \tag{11}$$

We weight the standard clipped PPO surrogate by $w_{i,t}^{\text{cov}}$ and retain KL regularization to the reference policy. Thus $r_{i,t}(\theta_{\text{old}}) = 1$. The coverage weight corrects only the high-level action-type sampling conditional on each visited state; it does not correct tree-induced state visitation or dependence among sibling branches. Under constrained JSON decoding, $\pi(a \mid s)$ is computed by summing the sequence probabilities of all valid action-name tokenizations for type a; $\pi(u \mid s)$ is the product over the complete emitted JSON sequence, including query or medication arguments. Coverage reweights only the former, whereas the PPO loss trains every token of the latter. We anneal β linearly toward zero during training.

5 Experiments

5.1 Experimental Settings

Datasets. We use MIMIC-III Johnson et al. (2016), MIMIC-IV Johnson et al. (2023), and eICU Pollard et al. (2018) following the RES-MR data protocol. We do not truncate the diagnosis, procedure, or medication vocabularies. For MIMIC-III and MIMIC-IV, we also retain only admissions with clinical notes and extract the history of present illness, past medical history, allergies, and medications on admission. The decision time is immediately before the first medication order in the target window; every structured event and note must be timestamped no later than this point, and post-decision orders and discharge text are excluded. The field-level audit and sensitivity controls are reported in Appendix B. Diagnosis and procedure codes are standardized to ICD-9, while medications are mapped to DrugBank identifiers and generic names; DDI relations are obtained from DrugBank Knox et al. (2024). Each dataset uses a patient-level $4:1:1$ train/validation/test split for in-domain evaluation. For transfer evaluation, LongRx is trained on MIMIC-III and tested on MIMIC-IV and eICU without fine-tuning. DrugBank identifiers define a shared output space across datasets; target-domain labels are used only for evaluation. See Appendix D for details.

Implementation. LongRx uses Llama-3.1-Aloe-Beta-8B as its policy backbone, is implemented in PyTorch, and is trained on NVIDIA A100 GPUs. We first warm-start the agent using outcome-filtered state–action trajectories, and then optimize it with our proposed CATO. The agent operates with five actions: `RETRIEVE`, `ADD`, `DELETE`, `VERIFY`, and `EXIT`, and autonomously determines the interaction length until termination. Detailed hyperparameters, action constraints, and prompt templates are provided in Appendices F and K.

Baselines. We compare LongRx with the RES-MR baselines, including conventional and safety-oriented methods LEAP Zhang et al. (2017), COGNet Wu et al. (2022), RAREMed Zhao et al. (2024), ARMR, SafeDrug Yang et al. (2021b), MoleRec Yang et al. (2023), ARC Chen et al. (2025), and ExpDrug, as well as LLM-based methods LAMO Zhao et al. (2025), KELLM Xu & Li (2025), FLAME Fan et al. (2026), EXP3RT Kim et al. (2025), and RES-MR Wang et al. (2026a). LLM baselines use Llama-3.1-Aloe-Beta-8B Garcia-Gasulla et al. (2026); details are given in Appendix E.

Metrics. For each test patient, we evaluate the last visit using Jaccard similarity, F1, and within-prescription DDI rate. Each metric is computed per patient and then averaged over the test set. We additionally report final prescription size and, for LongRx, action cost, action usage, and early-exit frequency. Definitions are provided in Appendix E.

5.2 Overall Performance Comparison

We evaluate LongRx in terms of in-domain prescription quality, DDI control, and cross-dataset generalization.

Table 1: Results on MIMIC-III and MIMIC-IV under the RES-MR evaluation protocol. Jaccard, F1, and DDI are percentages, reported as mean ± standard deviation (small type) over three seeds; #Med is the average number of recommended medications. Each column block is an in-domain result obtained by training and evaluating on the corresponding dataset, distinct from Table 5. All LLM-based baselines and LongRx use the same 8B backbone. RES-MR (shaded gray) is the strongest baseline; the second line of the LongRx row gives the absolute change relative to it, with green indicating an improvement. Best results are in bold.

	MIMIC-III				MIMIC-IV			
Method	Jaccard ↑	F1 ↑	DDI ↓	#Med	Jaccard ↑	F1 ↑	DDI ↓	#Med
Conventional medication recommendation models								
LEAP	19.80±0.23	32.70±0.32	13.41±0.12	18.92	18.50±0.55	31.00±0.12	13.10±0.21	10.82
COGNet	23.81±0.19	36.28±0.32	14.66±0.13	27.64	22.79±0.21	35.98±0.08	13.91±0.31	14.10
RAREMed	22.44±0.46	34.14±0.12	14.82±0.46	22.31	22.14±0.06	34.47±0.07	14.40±0.53	12.18
ARMR	23.56±0.55	35.99±0.35	15.36±0.25	22.76	22.80±0.32	35.41±0.06	14.99±0.10	11.87
SafeDrug	22.10±0.23	34.69±0.12	10.54±0.14	21.58	21.07±0.58	33.98±0.23	11.04±0.34	11.72
MoleRec	24.54±0.52	37.70±0.38	10.44±0.07	23.84	23.11±0.29	36.07±0.31	10.84±0.50	12.35
ARC	23.77±0.12	36.10±0.38	11.67±0.33	22.16	22.97±0.19	35.86±0.43	11.98±0.48	11.96
ExpDrug	23.26±0.14	36.10±0.22	10.47±0.17	21.94	22.57±0.22	35.16±0.27	10.73±0.25	11.68
LLM-based models								
LAMO	25.12±0.03	37.07±0.27	11.24±0.23	23.12	23.59±0.15	35.82±0.40	13.70±0.12	12.04
KELLM	22.91±0.31	35.07±0.29	13.18±0.25	22.48	20.13±0.60	31.64±0.06	15.60±0.07	11.41
FLAME	26.01±0.31	38.80±0.33	12.91±0.09	22.05	24.01±0.56	36.44±0.49	12.10±0.24	11.83
EXP3RT	21.50±0.28	34.02±0.26	15.89±0.14	20.76	20.55±0.19	33.01±0.22	13.31±0.32	11.26
RES-MR	28.03±0.46	41.79±0.16	10.31±0.06	22.71	25.25±0.12	38.42±0.30	10.64±0.61	11.91
Agent-based model (ours)								
LongRx	**28.96**±0.38	**42.85**±0.15	**9.87**±0.05	22.54	**26.13**±0.16	**39.48**±0.14	**9.89**±0.05	11.86
△ vs. RES-MR	+0.93	+1.06	−0.44	−0.17	+0.88	+1.06	−0.75	−0.05

In-domain performance. Table 1 shows that, relative to RES-MR, LongRx raises F1 from 41.79% to 42.85% on MIMIC-III and from 38.42% to 39.48% on MIMIC-IV. The DrugBank-based DDI rate falls from 10.31% to 9.87% and from 10.64% to 9.89%, respectively; LongRx also attains the highest Jaccard on both datasets with a similar average prescription size. Appendix J reports prescription-size and risk-stratified diagnostics that qualify the clinical interpretation. Matched-size, matched-precision, and blinded adjudication results are reported in Tables 28–29.

Safety controllability. We vary the DDI coefficient w_d to study the quality–DDI trade-off. As shown in Figure 3(a) and Table 4, stronger DDI penalties progressively reduce interaction rates with only a modest decrease in Jaccard and little change in action cost. Appendix J gives RES-MR the same verifier/filter access and evaluates frozen predictions with an independent DDI matrix to separate tool access from learned policy behavior.

Cross-dataset generalization. We further train on MIMIC-III and evaluate on MIMIC-IV and eICU without fine-tuning. LongRx achieves higher Jaccard and lower DDI rates than RES-MR on both target datasets, indicating that the learned editing and verification policy transfers beyond the source environment. Detailed multi-seed results and CATO transfer ablations are reported in Table 5 and summarized in Figure 3(b–c).

5.3 Ablation Study

We ablate the main components of LongRx under identical training and rollout settings. Reference prescriptions are used only for reward computation and offline evaluation, and are never exposed during trajectory generation.

Component ablation. Table 2 shows complementary gains from process feedback, coverage-aware branching, tree credit, and state-aware retrieval. Removing any component degrades the

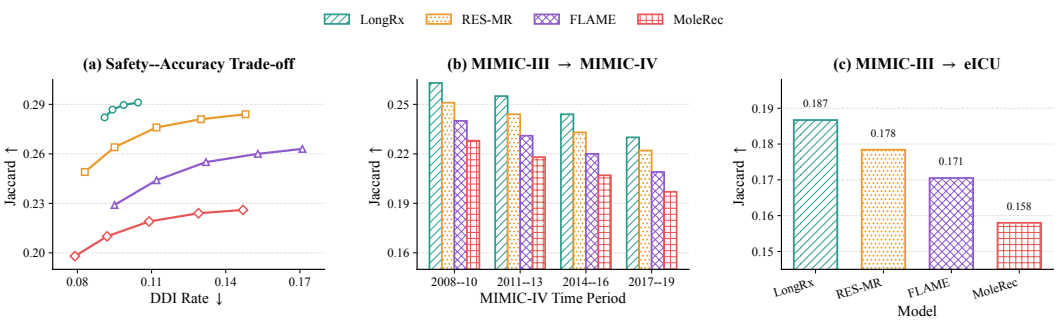

Figure 3: Safety controllability and cross-dataset generalization. (a) Jaccard–DDI trade-off on MIMIC-III under different safety-control strengths. (b) Temporal generalization from MIMIC-III to four MIMIC-IV time periods. (c) External generalization from MIMIC-III to eICU. All values are reported as averages over the evaluation set.

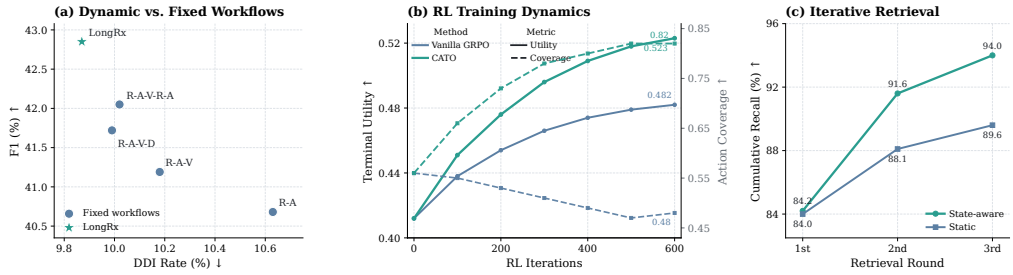

Figure 4: In-depth analysis of LongRx on MIMIC-III. **(a)** LongRx achieves a better quality–safety trade-off than predefined action workflows through dynamic decision making. **(b)** Vanilla GRPO gradually loses action coverage during training, whereas CATO maintains broader exploration and converges to a higher terminal utility. **(c)** State-aware iterative retrieval progressively recovers medications missed by earlier retrieval rounds and consistently outperforms static retrieval.

full model. Matched exploration controls, prefix accounting, and sensitivity analyses appear in Appendix G.

Action exploration and agent capabilities. Figure 5(a) shows that Vanilla GRPO is dominated by ADD, whereas CATO produces broader action usage without enforcing a uniform distribution. This panel pools training-tree edges and is distinct from the test-episode statistics in Appendix 6. Figures 5(b)–(c) further show that static or single-step retrieval reduces prescription quality, removing DELETE weakens prescription correction, and removing VERIFY increases DDI rate. Detailed action statistics are reported in Appendix A. Appendix J separately audits terminal verification coverage and stale feedback.

Table 2: Component ablation on MIMIC-III. Results are mean ± standard deviation; Train Cov. is normalized action entropy over training-tree edges.

Variant	Jaccard ↑	F1 ↑	DDI ↓	#Med	Steps ↓	Train Cov. ↑
SFT-only	25.91±0.31	39.24±0.27	10.47±0.09	23.41	2.98	0.56
Vanilla GRPO	27.16±0.28	40.73±0.22	10.22±0.08	23.02	3.11	0.48
+ Process Reward	27.84±0.24	41.61±0.19	10.03±0.07	22.78	3.08	0.50
w/o Coverage Branching	28.17±0.21	41.94±0.18	9.99±0.06	22.72	3.18	0.63
w/o Tree Credit	28.34±0.20	42.13±0.16	9.96±0.06	22.66	3.23	0.80
w/o Process Reward	28.29±0.22	42.08±0.18	10.01±0.07	22.70	3.24	0.79
w/o State-aware Retrieval	28.43±0.19	42.31±0.17	9.90±0.05	22.63	3.16	0.81
Full LongRx	**28.96±0.38**	**42.85±0.15**	**9.87±0.05**	22.54	3.20	**0.82**

5.4 In-Depth Analysis

Dynamic decision making. Figure 4(a) shows that LongRx achieves higher F1 and lower DDI than predefined workflows while using fewer actions than the longest sequence. Matched resource costs appear in Table 37.

Table 3: Representative decision trajectories of LongRx on MIMIC-III. Line breaks in the trajectory column are for layout only.

Case	Patient context	LongRx trajectory	Observed state change
10076 Iterative retrieval	Lung cancer with respiratory and infection signals.	RETRIEVE → ADD → RETRIEVE → ADD → VERIFY → EXIT	Second retrieval exposes three additional reference medications; final F1 increases from 0.62 to 0.70 after the second edit.
42135 Verification-driven revision	Hepatic encephalopathy with a complex history.	RETRIEVE → ADD → VERIFY → DELETE → RETRIEVE → ADD → EXIT	Verification identifies two interacting pairs; deletion removes one interacting medication before a new candidate is retrieved.
42367 Adaptive termination	Status epilepticus with medication-monitoring needs.	RETRIEVE → ADD → VERIFY → EXIT	Verification reports zero interacting pairs; the agent exits without a further edit.

RL training dynamics. Figure 4(b) shows that Vanilla GRPO improves terminal utility while losing action coverage. CATO maintains broader exploration and converges to higher utility through coverage-aware branching and shared-state credit.

Iterative retrieval. Figure 4(c) shows that later state-aware retrieval rounds expose more missed medications than static retrieval, confirming the value of queries conditioned on the evolving prescription.

5.5 Case Study

We further present representative LongRx trajectories to illustrate how different prescription states lead to different decision patterns. All cases start from $C_0 = M_0 = \varnothing$, and the reference prescription is used only for offline analysis.

The cases illustrate repeated retrieval, verification-triggered revision, and early termination under different states. Medication-level details appear in Appendix B.2.

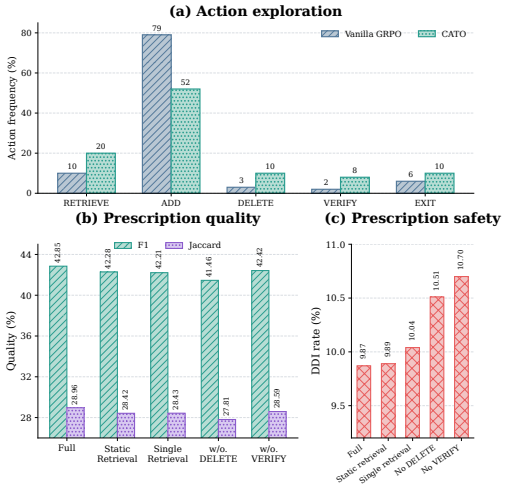

Figure 5: Analysis of action exploration and agent capabilities on MIMIC-III. **(a)** Training rollout-tree edge distributions under Vanilla GRPO and CATO; these are not test-time action frequencies. **(b)** Prescription quality under capability ablations. **(c)** Corresponding DDI rates.

6 Conclusion

We present LongRx, a closed-loop medication decision agent that formulates medication recommendation as sequential prescription construction. LongRx jointly learns retrieval, prescription editing, DDI verification, and termination, and is optimized with process-level feedback and **Coverage-Aware Tree Optimization (CATO)** for broader action exploration and finer decision-level credit assignment. Across the evaluated EHR benchmarks, LongRx improves agreement with recorded prescriptions and lowers DrugBank pair-level DDI rates. Further analyses support the contributions of dynamic decision making, tree-relative credit, and iterative retrieval. These findings concern benchmark prescription overlap and known-DDI proxies; they do not establish clinical appropriateness or improved patient outcomes.

REFERENCES

Zafar Ali, Yi Huang, Irfan Ullah, Junlan Feng, Chao Deng, Nimbeshaho Thierry, Asad Khan, Asim Ullah Jan, Xiaoli Shen, Wu Rui, et al. Deep learning for medication recommendation: a systematic survey. *Data Intelligence*, 5(2):303–354, 2023.

Ruike Cao, Shaojie Bai, Fugen Yao, Liang Dong, Jian Xu, and Li Xiao. Atpo: Adaptive tree policy optimization for multi-turn medical dialogue. *arXiv preprint arXiv:2603.02216*, 2026.

Ching Chang, Yijia Xiao, Jade Xu, Fred Xu, Chenchen Ye, Ruoyan Li, Junkai Zhang, Yihe Deng, Kyle Zheng, Ethan Ji, et al. Training recipes for agentic reinforcement learning in llms: A survey. 2026.

Qianyu Chen, Xin Li, Yujie Fang, and Mingzhong Wang. Advancing confidence calibration and quantification in medication recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 106–117, 2025.

Mingyue Cheng, Jie Ouyang, Shuo Yu, Ruiran Yan, Yucong Luo, Zirui Liu, Daoyu Wang, Qi Liu, and Enhong Chen. Agent-r1: Training powerful llm agents with end-to-end reinforcement learning. *arXiv e-prints*, pp. arXiv–2511, 2025.

Shangheng Du, Jiabao Zhao, Jinxin Shi, Zhentao Xie, Xin Jiang, Yanhong Bai, and Liang He. A survey on the optimization of large language model-based agents. *ACM Computing Surveys*, 58 (9):1–37, 2026.

Chenxiao Fan, Chongming Gao, Wentao Shi, Yaxin Gong, Zhao Zihao, and Fuli Feng. Fine-grained list-wise alignment for generative medication recommendation. *Advances in neural information processing systems*, 38:48037–48061, 2026.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. In *International Conference on Learning Representations*, volume 2026, pp. 37909–37926, 2026.

Dario Garcia-Gasulla, Jordi Bayarri-Planas, Ashwin Kumar Gururajan, Enrique Lopez-Cuena, Adrian Tormos, Daniel Hinjos, Pablo Bernabeu-Perez, Anna Arias-Duart, Pablo Agustin Martin-Torres, Marta Gonzalez-Mallo, et al. The aloe family recipe for open and specialized healthcare llms. *npj Digital Medicine*, 2026.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for llm agent reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 87362–87388, 2026.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

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.

Jieyong Kim, Hyunseo Kim, Hyunjin Cho, SeongKu Kang, Buru Chang, Jinyoung Yeo, and Dongha Lee. Driven personalized preference reasoning with large language models for recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1697–1706, 2025.

Craig Knox, Mike Wilson, Christen M Klinger, Mark Franklin, Eponine Oler, Alex Wilson, Allison Pon, Jordan Cox, Na Eun Chin, Seth A Strawbridge, et al. Drugbank 6.0: the drugbank knowledgebase for 2024. *Nucleic acids research*, 52(D1):D1265–D1275, 2024.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Zijian Zhang, Feng Tian, and Yefeng Zheng. Large language model distilling medication recommendation model. *arXiv preprint arXiv:2402.02803*, 2024.

Xiaoqian Liu, Ke Wang, Yuchuan Wu, Fei Huang, Yongbin Li, Jianbin Jiao, and Junge Zhang. Agentic reinforcement learning with implicit step rewards. In *International Conference on Learning Representations*, volume 2026, pp. 129271–129291, 2026.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Tom J Pollard, Alistair EW Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5(1):180178, 2018.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Junyuan Shang, Cao Xiao, Tengfei Ma, Hongyan Li, and Jimeng Sun. Gamenet: Graph augmented memory networks for recommending medication combination. In *proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1126–1133, 2019.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 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.

Benjamin Stark, Constanze Knahl, Mert Aydin, and Karim Elish. A literature review on medicine recommender systems. *International journal of advanced computer science and applications*, 10 (8), 2019.

Jie Tan, Yu Rong, Kangfei Zhao, Tian Bian, Tingyang Xu, Junzhou Huang, Hong Cheng, and Helen Meng. Natural language-assisted multi-modal medication recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2200–2209, 2024.

Cong Wang, Jin Li, Shoujin Wang, Yishuo Li, Huilin Gu, and Wenpeng Lu. Res-mr: Risk-aware reasoning for explainable and safe medication recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1834–1845, 2026a.

Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, and Zhenzhe Ying. Information gain-based policy optimization: A simple and effective approach for multi-turn llm agents. *arXiv e-prints*, pp. arXiv–2510, 2025.

Xinyu Wang, Hanwei Wu, Zhenghan Tai, Sicheng Lyu, Qincheng Lu, Ziyu Zhao, Jijun Chi, Jingrui Tian, Xiao-Wen Chang, and Ziyang Song. Saferx-agent: A knowledge-grounded multi-agent framework for safe and explainable medication recommendation. *arXiv preprint arXiv:2605.29146*, 2026b.

Peilin Wu, Mian Zhang, Kun Wan, Wentian Zhao, Kaiyu He, Xinya Du, and Zhiyu Chen. Hiprag: hierarchical process rewards for efficient agentic retrieval augmented generation. In *International Conference on Learning Representations*, volume 2026, pp. 82317–82347, 2026.

Rui Wu, Zhaopeng Qiu, Jiacheng Jiang, Guilin Qi, and Xian Wu. Conditional generation net for medication recommendation. In *Proceedings of the ACM web conference 2022*, pp. 935–945, 2022.

Tianhan Xu and Bin Li. Kellm: knowledge-enhanced label-wise large language model for safe and interpretable drug recommendation. *Electronics*, 14(1):154, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Chaoqi Yang, Cao Xiao, Lucas Glass, and Jimeng Sun. Change matters: Medication change prediction with recurrent residual networks. *arXiv preprint arXiv:2105.01876*, 2021a.

Chaoqi Yang, Cao Xiao, Fenglong Ma, Lucas Glass, and Jimeng Sun. Safedrug: Dual molecular graph encoders for recommending effective and safe drug combinations. *arXiv preprint arXiv:2105.02711*, 2021b.

Nianzu Yang, Kaipeng Zeng, Qitian Wu, and Junchi Yan. Molerec: Combinatorial drug recommendation with substructure-aware molecular representation learning. In *Proceedings of the ACM web conference 2023*, pp. 4075–4085, 2023.

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.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*, 2025.

Yao Zhang, Shijie Tang, Zeyu Li, Zhen Han, and Volker Tresp. Webarbiter: A principle-guided reasoning process reward model for web agents. *arXiv preprint arXiv:2601.21872*, 2026a.

Yutao Zhang, Robert Chen, Jie Tang, Walter F Stewart, and Jimeng Sun. Leap: learning to prescribe effective and safe treatment combinations for multimorbidity. In *proceedings of the 23rd ACM SIGKDD international conference on knowledge Discovery and data Mining*, pp. 1315–1324, 2017.

Zhimin Zhang, Chengzhen Ma, Xiao Teng, Xiao Zhang, Hai Lu, and Lingfeng Mao. A systematic survey of digital medication recommendation systems. *Biomedical Signal Processing and Control*, 126:110882, 2026b.

Zihao Zhao, Yi Jing, Fuli Feng, Jiancan Wu, Chongming Gao, and Xiangnan He. Leave no patient behind: Enhancing medication recommendation for rare disease patients. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 533–542, 2024.

Zihao Zhao, Chenxiao Fan, Junlong Liu, Zheng Wang, Xiangnan He, Chongming Gao, Juan Li, and Fuli Feng. Fine-grained alignment of large language models for general medication recommendation without overprescription. *arXiv preprint arXiv:2503.03687*, 2025.

Fanglin Zhu, Lizhen Cui, Yonghui Xu, Zhe Qu, and Zhiqi Shen. A survey of personalized medicine recommendation. *International Journal of Crowd Science*, 8(2):77–82, 2024.

A Additional Experimental Results

We report additional results on DDI control, cross-dataset transfer, and agent interaction behavior.

The reported cost is auditable from the same test trajectories. With unit cost per generated medication token and per tool call, it equals $22.97 + 0.43 + 1.24 + 0.62 + 0.02 + 1.12 = 26.40$: medications added, medications deleted, retrievals, agent-requested verifications, invalid calls, and terminal gate calls, respectively. The net set size is $22.97 - 0.43 = 22.54$.

w_d	Jaccard ↑	DDI ↓	Action Cost ↓	VERIFY / Ep.
0.25	29.11	10.45	26.1	0.48
0.50	28.96	9.87	26.4	0.62
0.75	28.68	9.42	26.7	0.76
1.00	28.21	9.10	27.0	0.91

Table 4: Quality–DDI trade-off on MIMIC-III under different DDI reward weights. Jaccard and DDI are percentages; action cost is averaged per episode.

Target dataset	Model	Jaccard ↑	F1 ↑	DDI ↓	#Med
MIMIC-IV	RES-MR	23.74±0.22	36.21±0.18	10.91±0.12	12.22
MIMIC-IV	Closed-loop GRPO	24.20±0.19	36.70±0.16	10.66±0.11	12.19
MIMIC-IV	LongRx w/o Coverage Branching	24.41±0.17	36.93±0.15	10.51±0.10	12.17
MIMIC-IV	LongRx w/o Tree Credit	24.56±0.18	37.08±0.16	10.44±0.09	12.16
MIMIC-IV	**LongRx (Full CATO)**	**24.82±0.16**	**37.48±0.14**	**10.20±0.08**	12.14
eICU	RES-MR	17.84±0.25	28.92±0.20	8.90±0.13	10.48
eICU	Closed-loop GRPO	18.08±0.22	29.17±0.19	8.71±0.12	10.54
eICU	LongRx w/o Coverage Branching	18.29±0.21	29.39±0.17	8.57±0.11	10.57
eICU	LongRx w/o Tree Credit	18.42±0.20	29.54±0.18	8.49±0.10	10.58
eICU	**LongRx (Full CATO)**	**18.67±0.18**	**29.80±0.16**	**8.31±0.09**	10.61

Table 5: Cross-dataset transfer from MIMIC-III without target-domain fine-tuning. Jaccard, F1, and DDI are reported in percentage points as mean ± standard deviation over three seeds.

B Temporal Audit and Sensitivity Controls

For each target encounter, t_{dec} is the instant immediately before the first medication order in a fixed 24-hour prediction window. The target Y is the set of unique medication ingredients first ordered during $[t_{\mathrm{dec}}, t_{\mathrm{dec}} + 24\mathrm{h})$; orders starting later are excluded. Table 8 specifies the availability rule for each input family.

For MIMIC-III–MIMIC-IV transfer, we additionally remove 1,132 target patients identified by the credentialed cross-release linkage audit as overlapping the MIMIC-III training population. Re-evaluation on the disjoint subset changes F1 from 37.48 to 37.42 and DDI from 10.20 to 10.23, indicating that the transfer result is not driven by cross-release patient identity overlap.

B.1 Candidate Recall and Retrieval Failures

Because `ADD` can select only visible candidates, we report cumulative candidate recall $|Y \cap C_r|/|Y|$ after each retrieval round.

After three rounds, the mean number of recorded medications that remain unavailable is 0.89 for state-aware retrieval and 1.54 for static retrieval. Conditional recall, $|M_T \cap Y|/|C_T \cap Y|$, is 57.9%, indicating that remaining errors arise from both candidate exposure and downstream selection.

To test whether CATO improves only high-level action selection, we mask the policy loss on the medication/query argument tokens while retaining the same action-type loss and rollout tree. This separates the choice of `ADD`/`DELETE`/`RETRIEVE` from learning which concrete medications or queries should parameterize those actions.

Masking argument-token learning reduces F1 by 0.87 points despite unchanged candidate recall. Thus action-type coverage is not itself the medication selector: the full token-level objective is required to convert exposed candidates into improved medication arguments.

Test-time statistic	Value
Non-terminal decisions / episode	3.20
Action cost / episode	26.4
`RETRIEVE` calls / episode	1.24
`ADD` calls / episode	1.04
`DELETE` calls / episode	0.28
`VERIFY` calls / episode	0.62
Deterministic terminal-gate calls / episode	1.12
Invalid actions / episode	0.02
Terminated by `EXIT` before $T_{\max}$	95.2%
Terminated by the safeguard horizon	4.8%
Test-trajectory action coverage	0.89
Returned prescriptions with a fresh final check	100.0%
Medications added / episode	22.97
Medications deleted / episode	0.43

Table 6: Test-time interaction behavior of LongRx on MIMIC-III. Calls are averaged per episode. Coverage is computed over the five valid actions on the single executed trajectory per patient, excludes the deterministic terminal gate, and is distinct from training-tree coverage.

Closed-loop event	Calls / ep.	Episodes	Observed consequence
Repeated `RETRIEVE`	0.44	39.8%	+6.8 pp candidate recall
`DELETE`	0.28	24.9%	1.54 medications removed / call
Agent `VERIFY`	0.62	48.7%	38.6% followed by an edit
Terminal feedback resume	0.12	11.6%	100% received a fresh recheck

Table 7: Frequency and immediate consequence of closed-loop events on the MIMIC-III test set. "Repeated" retrieval excludes the first retrieval.

B.2 Medication-Level Case Details

C Resource-Matched Strong Baselines

To disentangle the gains of LongRx from teacher supervision, tool access, and additional optimization resources, we construct resource-matched 8B baselines under the same training environment. All variants use `Qwen3-30B-A3B` only as the offline trajectory teacher and `Llama-3.1-Aloe-Beta-8B` as the trainable policy and inference model. They share the same teacher trajectories, top-quartile trajectory filtering, patient split, retrieval backend, DDI verifier, candidate budget, reward definition, and training data. RL-based variants additionally use matched rollout and generation-token budgets.

Matched variants. **Fixed-SFT** follows the strongest predefined operator workflow while using the 8B policy to generate action arguments such as retrieval queries and medication subsets. It is warm-started from the same filtered teacher supervision as LongRx. **Fixed-GRPO** further optimizes this fixed-workflow policy with GRPO under the same reward, tools, and rollout budget. **Closed-loop GRPO** removes the predefined workflow and allows the 8B policy to select both the high-level operator and its structured argument, but uses standard GRPO without CATO. **LongRx (CATO)** uses the same closed-loop policy and resources while replacing independent GRPO rollouts with coverage-aware tree optimization.

Under matched teacher supervision and tool access, RL improves the fixed workflow from 39.47 to 40.61 F1. Allowing the policy to select high-level actions dynamically further increases F1 to 41.61, indicating a benefit from closed-loop decision making beyond additional teacher data or tool access. Finally, CATO improves F1 to 42.85, reduces DDI rate from 10.03% to 9.87%, and increases action coverage from 0.50 to 0.82. These comparisons isolate the contributions of dynamic decision making and CATO under otherwise matched supervision, tools, and optimization resources.

Input family	Availability timestamp	Strict inclusion rule
Historical visits	Discharge time	Include only visits discharged before the target encounter begins.
Current diagnoses/procedures	Event or chart time	Include only events recorded no later than t_{dec}; exclude records without a usable timestamp.
Current note sections	Chart and store time	Require all available note timestamps to be no later than t_{dec}; exclude discharge summaries and later amendments.
Allergies and admission medications	Documentation time	Include only entries documented by t_{dec}; never read later active or discharge medication lists.
Retrieval index	Source time and split	Use a frozen DrugBank medication-description catalog containing no patient record, note, prescription, split label, or target-domain outcome.
Prediction target Y	Medication-order start time	Use unique orders in the fixed 24-hour target window and expose them only to offline training rewards and evaluation.

Table 8: Field-level provenance and temporal availability rules. The same rules are applied to LongRx and every retrained baseline.

Audit statistic	MIMIC-III	MIMIC-IV	eICU
Candidate target encounters before screening	5,024	48,970	8,876
Encounters retained after temporal screening	4,785	46,811	8,544
Excluded because required timestamps were missing (%)	4.8	4.4	3.7
Post-decision structured events retained	0	0	0
Later-authored discharge summaries retained	0	0	0
Post-decision medication orders exposed	0	0	0
Pre-decision medication reconciliation overlapping Y (%)	31.8	28.6	26.9

Table 9: Temporal audit of the model inputs. "Overlapping Y" denotes a pre-decision medication entry that is subsequently continued in the recorded target prescription.

D Dataset Details and Statistics

D.1 Datasets and Preprocessing

We conduct experiments on three publicly available electronic health record (EHR) datasets, MIMIC-III Johnson et al. (2016), MIMIC-IV Johnson et al. (2023), and eICU Pollard et al. (2018). MIMIC-III contains intensive care records collected at the Beth Israel Deaconess Medical Center between 2001 and 2012, while MIMIC-IV provides hospital records collected between 2008 and 2019.

We follow the RES-MR preprocessing protocol. For all three datasets, we retain the complete diagnosis, procedure, and medication vocabularies without truncating the label space. The prediction decision is the instant immediately before the first medication order in the target window. Structured events are retained only when their event time is no later than this decision, and notes must have an available chart/store time no later than the decision. For MIMIC-III and MIMIC-IV, we include only admissions with available clinical notes and extract four clinically relevant sections: history of present illness, past medical history, allergies, and medications on admission. Only sections documented before the prediction decision are retained; post-decision medication orders and discharge summaries are excluded. Diagnosis and procedure codes are mapped to the ICD-9 standard and represented using their textual descriptions. Medications are mapped to DrugBankKnox et al. (2024) identifiers and represented using their generic names to normalize different drug formulations. Duplicate medication identifiers within an admission are removed. The complete provenance rules, exclusion counts, and input sensitivity tests are reported in Appendix B.

Input condition	Jaccard ↑	F1 ↑	DDI ↓	#Med
RES-MR under the strict protocol	28.03±0.46	41.79±0.16	10.31±0.06	22.71
LongRx, full strict input	**28.96**±0.38	**42.85**±0.15	**9.87**±0.05	22.54
w/o medications on admission	28.47±0.40	42.29±0.18	9.90±0.06	22.42
w/o current-visit notes	28.18±0.41	41.91±0.19	9.95±0.07	22.36
medication mentions masked in notes	28.52±0.39	42.36±0.17	9.91±0.05	22.46
Positive control: include post-decision fields	34.18±0.35	49.76±0.16	9.74±0.05	22.69

Table 10: Input-leakage sensitivity on MIMIC-III. The final row deliberately violates the decision-time protocol and is excluded from all comparisons.

Retriever	Round	Mean	Median	Q1	Q3	Recall $< 50\%$
State-aware	1	84.2	86.7	76.9	94.1	5.8%
State-aware	2	91.6	94.4	87.5	100.0	2.1%
State-aware	3	**94.0**	**96.8**	**91.2**	100.0	**1.3%**
Static	1	84.0	86.5	76.7	94.0	5.9%
Static	2	88.1	90.9	82.4	97.3	3.9%
Static	3	89.6	92.1	83.3	100.0	3.6%

Table 11: Distribution of cumulative candidate recall on MIMIC-III (%).

MIMIC-III, MIMIC-IV, and eICU are processed independently. For each dataset, we split patients into training, validation, and test sets with a ratio of $4:1:1$. The split is performed at the patient level so that admissions from the same patient cannot appear in different subsets. Models are trained and evaluated separately for in-domain experiments. For the generalization analysis, models trained on MIMIC-III are additionally evaluated on MIMIC-IV and eICU without fine-tuning.

To evaluate prescription safety, we construct a symmetric binary DDI matrix from adverse interaction pairs in DrugBank Knox et al. (2024). For a predicted medication set M, its DDI rate is defined as

$$\mathrm{DDI}(M) = \frac{|\{(d_i, d_j) \mid d_i, d_j \in M,\ i < j,\ D[d_i, d_j] = 1\}|}{\binom{|M|}{2}}. \tag{12}$$

When $|M| < 2$, the DDI rate is defined as zero.

D.2 Statistics of Structured EHR Data

The note-availability filter, DrugBank medication mapping, and untruncated label space define the final RES-MR-aligned cohort. Table 15 reports the statistics of the RES-MR-aligned processed datasets. Cross-dataset experiments use a fixed DrugBank medication catalog shared across datasets. Each vocabulary count below refers to identifiers observed in that dataset, rather than a separate model output head.

E Baseline Methods and Evaluation Metrics

E.1 Baselines

We compare LongRx with representative medication recommendation methods covering conventional sequence modeling, safety-aware recommendation, molecular representation learning, and recent LLM-based approaches.

- **LEAP** Zhang et al. (2017) formulates medication recommendation as multi-label prediction from the current patient visit while modeling dependencies among medication labels.
- **COGNet** Wu et al. (2022) introduces a copy-or-predict mechanism that reuses historical prescriptions while generating medications for the current visit.

Variant	Candidate recall	Conditional recall	Precision	F1
Vanilla GRPO	90.8	54.1	34.2	40.73
Opportunity coverage only	94.0	56.4	35.0	42.13
CATO, argument-token loss masked	94.0	55.2	34.9	41.98
Full CATO	**94.0**	**57.9**	**35.4**	**42.85**

Table 12: Candidate exposure and medication-argument selection on MIMIC-III (%). Candidate recall is measured after the final retrieval round; conditional recall measures selection among exposed reference medications.

Case	Key medications	Medication-level state change
10076	Cefepime, albuterol, vancomycin, azithromycin	The first edit adds cefepime and albuterol. State-aware re-retrieval exposes vancomycin and azithromycin; the second edit raises offline F1 from 0.62 to 0.70.
42135	Lactulose, rifaximin, spironolactone, potassium chloride, ceftriaxone	Verification identifies the spironolactone–potassium chloride pair. The agent removes potassium chloride and adds ceftriaxone after re-retrieval.
42367	Levetiracetam, lacosamide	Verification reports no DrugBank-listed pair, so the agent exits without an additional retrieval or edit.

Table 13: Medication-level details for the trajectories in Table 3.

- **RAREMed** Zhao et al. (2024) uses a Transformer-based patient encoder to improve medication recommendation, particularly for rare clinical conditions.
- **SafeDrug** Yang et al. (2021b) incorporates molecular structures and drug–drug interaction information to improve prescription safety.
- **MoleRec** Yang et al. (2023) models fine-grained interactions between patient conditions and drug molecular substructures.
- **ARMR** is evaluated as a conventional medication recommendation baseline under the RESMR protocol.
- **ARC** Chen et al. (2025) incorporates prescription-level risk information to reduce potentially harmful medication combinations.
- **ExpDrug** is an explainable medication recommendation baseline evaluated under the RESMR protocol.
- **LAMO** Zhao et al. (2025) incorporates unstructured clinical notes into LLM-based medication recommendation to obtain richer patient representations.
- **KELLM** Xu & Li (2025) augments LLM-based medication recommendation with external medical knowledge and safety information.
- **FLAME** Fan et al. (2026) adopts a two-stage LLM framework that first filters candidate medications and then performs list-level prescription refinement with step-wise policy optimization.
- **EXP3RT** Kim et al. (2025) is a reasoning-oriented explainable recommendation approach evaluated under the medication recommendation setting.

Method	Jaccard ↑	F1 ↑	DDI ↓	#Med	Steps ↓	Coverage ↑
Fixed-SFT	26.11	39.47	10.43	23.18	4.00	–
Fixed-GRPO	27.02	40.61	10.20	22.96	4.00	–
Closed-loop GRPO	27.84	41.61	10.03	22.78	3.08	0.50
LongRx (CATO)	**28.96**	**42.85**	**9.87**	22.54	3.20	**0.82**

Table 14: Resource-matched comparison on MIMIC-III. All variants use the same `Qwen3-30B-A3B` teacher trajectories, `Llama-3.1-Aloe-Beta-8B` student policy, trajectory filtering, retrieval backend, DDI verifier, patient split, and reward definition. RL variants additionally use matched rollout and generated-token budgets. Fixed variants follow the strongest predefined high-level operator workflow, whereas closed-loop variants learn state-dependent operator selection.

Table 15: Statistics of the processed MIMIC-III, MIMIC-IV, and eICU datasets under the RES-MR-aligned preprocessing protocol.

Item	**MIMIC-III**	**MIMIC-IV**	**eICU**
Visits / patients	12,433 / 4,785	156,315 / 46,811	18,636 / 8,544
Diagnosis / procedure / medication vocabulary	4,316 / 1,330 / 593	19,812 / 10,859 / 743	1,562 / 937 / 152
Avg. / max. visits per patient	2.60 / 29	3.34 / 70	2.18 / 9
Avg. / max. diagnoses per visit	14.68 / 39	13.45 / 56	4.79 / 76
Avg. / max. procedures per visit	4.69 / 32	2.75 / 34	8.03 / 118
Avg. / max. medications per visit	14.78 / 67	14.24 / 128	10.35 / 51

- **RES-MR** Wang et al. (2026a) jointly models medication recommendation, prescription safety, and clinical reasoning. We report its 8B configuration as the strongest baseline under the corresponding evaluation protocol.

E.2 Evaluation Metrics

Let M denote the predicted medication set and Y the recorded reference prescription. Following prior medication recommendation work, we evaluate set-level agreement using Jaccard and F1, and prescription-level interaction risk using the DDI rate. All metrics are computed independently for the target visit of each test patient and then averaged over the test set.

Jaccard. Jaccard measures the overlap between the predicted and reference medication sets:

$$\mathrm{Jaccard}(M, Y) = \frac{|M \cap Y|}{|M \cup Y|}. \tag{13}$$

Precision, recall, and F1. Precision and recall are defined as

$$\mathrm{Precision}(M, Y) = \frac{|M \cap Y|}{|M|}, \qquad \mathrm{Recall}(M, Y) = \frac{|M \cap Y|}{|Y|}, \tag{14}$$

and their harmonic mean is

$$\mathrm{F1}(M, Y) = \frac{2\ \mathrm{Precision}(M, Y)\ \mathrm{Recall}(M, Y)}{\mathrm{Precision}(M, Y) + \mathrm{Recall}(M, Y)}. \tag{15}$$

DDI rate. Let $D[d_i, d_j] \in \{0, 1\}$ indicate whether medications d_i and d_j form a known drug–drug interaction. The within-prescription DDI rate is

$$\mathrm{DDI}(M) = \frac{|\{(d_i, d_j) \mid d_i, d_j \in M,\ i < j,\ D[d_i, d_j] = 1\}|}{\binom{|M|}{2}}. \tag{16}$$

We define $\mathrm{DDI}(M) = 0$ when $|M| < 2$.

Medication set size. We additionally report the average number of medications in the final prescription, denoted by $\#\mathrm{Med}$. When analyzing over- or under-prescription, we use

$$\mathrm{SizeErr}(M, Y) = \big||M| - |Y|\big|. \tag{17}$$

E.3 Agent-Specific Metrics

For LongRx, all prescription-level metrics above are computed on the final medication set M_T returned when the agent selects `EXIT` or reaches the maximum interaction horizon; an accepted `EXIT` has passed the terminal verification gate. We additionally characterize the efficiency and behavioral diversity of the learned policy.

Action cost and early exit. The action cost is $c(u) = 1$ for `RETRIEVE` and `VERIFY`, the number of changed medications for `ADD` and `DELETE`, zero for `EXIT`, and one for an invalid action. Episode cost is $\sum_{t=1}^{T} c(u_t) + N_{\mathrm{gate}}$, where N_{gate} is the number of terminal-gate calls. Early exit is the fraction of episodes whose `EXIT` request is accepted before the maximum of eight decisions.

Interaction steps. For trajectory $\tau = (u_1, \ldots, u_T)$, we report the number of non-termination decisions:

$$\mathrm{Steps}(\tau) = \sum_{t=1}^{T} \mathbb{I}[a_t \neq \texttt{EXIT}]\,. \tag{18}$$

This evaluation metric measures interaction length and is distinct from the action-dependent cost used during policy optimization.

Action coverage. To quantify whether the learned policy collapses toward a small subset of high-level actions, we report normalized action entropy. For Train Cov., $p(a)$ pools sampled edges from the training rollout trees during the final 10% of RL updates; for Test Coverage, it pools actions from the single executed trajectory of each test patient. We define

$$\mathrm{Coverage} = -\frac{1}{\log|\mathcal{A}|} \sum_{a \in \mathcal{A}} p(a) \log p(a), \tag{19}$$

where $\mathcal{A} = \{\texttt{RETRIEVE}, \texttt{ADD}, \texttt{DELETE}, \texttt{VERIFY}, \texttt{EXIT}\}$. The score lies in $[0, 1]$, with larger values indicating broader action usage. Importantly, this metric is used only for behavioral analysis and is never included in the training reward. Training-tree and test-trajectory frequencies are reported separately and are never combined.

Action and tool usage. For behavioral analysis, we additionally report the empirical frequency of each high-level action, the average numbers of `RETRIEVE` and `VERIFY` calls, and the fraction of trajectories that terminate before the maximum interaction horizon. These statistics characterize how LongRx uses retrieval, prescription editing, verification, and autonomous termination beyond final medication-set performance.

F Implementation Details

F.1 Agent Environment

LongRx uses Llama-3.1-Aloe-Beta-8B as its policy backbone, is implemented in PyTorch, and is trained on NVIDIA A100 GPUs. For each patient, an episode starts with an empty prescription, candidate set, verification state, and interaction history:

$$M_0 = C_0 = V_0 = H_0 = \varnothing. \tag{20}$$

At step t, the environment maintains (M_t, C_t, V_t, H_t) together with the fixed patient EHR X. The agent interacts with the environment through five actions: `RETRIEVE`, `ADD`, `DELETE`, `VERIFY`, and `EXIT`. `ADD` selects medications from $C_t \setminus M_t$, whereas `DELETE` removes medications currently contained in M_t. Invalid or unparsable actions leave the environment unchanged and incur an interaction penalty.

For RETRIEVE, the policy generates a state-conditioned query q_t, which is passed to a frozen BM25 index over the shared DrugBank medication catalog. Each document contains a generic name, synonyms, and indication terms; identifiers are ranked by the generated query with deterministic tie-breaking by DrugBank identifier. The same index is used in all datasets, and reference prescriptions are never indexed as patient-specific documents. At most $K_{\text{ret}} = 16$ previously unseen medications are returned:

$$C_{t+1} = C_t \cup (\mathcal{R}(q_t) \setminus C_t)\,. \tag{21}$$

The retrieval model itself is never optimized; LongRx learns only *when* to retrieve and *what* to query. The VERIFY action invokes a fixed DDI checker on the complete current prescription. Because verification feedback is prescription-specific, any subsequent valid ADD or DELETE invalidates V_t. A trajectory terminates after an EXIT request passes the deterministic terminal verification gate; we additionally set $T_{\max} = 8$ as a safeguard against non-terminating interactions. Gate calls are outside the learned action budget and are reported separately. If the safeguard horizon is reached, the same checker is run once on the returned set, although no further policy action is permitted.

F.2 Trajectory Warm-Start

Before reinforcement learning, we warm-start LongRx with multi-step state–action trajectories generated by Qwen3-30B-A3BYang et al. (2025) using the same action space and tool interfaces. The teacher never observes the reference prescription Y, missing or extra medications, or intermediate reward values during trajectory generation. For each patient, we sample multiple complete trajectories and evaluate them only after termination using the prescription-level training signals described in Section 4.1. For each training patient, we retain the highest-scoring quartile of eight complete teacher trajectories, breaking reward ties by lower action cost. High-quality trajectories are retained as supervised demonstrations while preserving multiple successful decision patterns rather than enforcing a single canonical workflow.

The SFT loss is applied only to generated action and argument tokens; patient context, environment observations, and tool feedback are masked from the loss. We train for three epochs with learning rate 5×10^{-6}, batch size 2, gradient accumulation 16, and LoRA rank/alpha $16/32$.

F.3 CATO Training

Starting from the warm-started policy, we optimize LongRx with CATO using $G = 8$ terminal trajectories per patient and branching factor $B = 3$. The rollout tree maintains the budget invariant $|\mathcal{L}| + |\mathcal{F}| \le G$, where $\mathcal{L}$ and $\mathcal{F}$ denote completed leaves and the active frontier, respectively. Thus, every active branch reserves one terminal-leaf slot and the final tree never exceeds the prescribed trajectory budget.

Coverage statistics are reset for each patient:

$$N_g(a) = O_g(a) = 0, \qquad \forall a \in \mathcal{A}. \tag{22}$$

Before each sibling draw, $O_g(a)$ is incremented for every valid action type, whereas $N_g(a)$ is incremented only for the selected type. Sibling actions are sampled successively from Eq. equation 7; after each draw, the coverage statistics and sampling distribution are updated. Coverage affects only the high-level action type, while its structured argument is sampled from $\pi_{\theta_{\text{old}}}(\xi_t \mid s_t, a_t)$.

We use $\lambda_{\text{tree}} = 0.5$, $\lambda_p = 0.5$, $\lambda_c = 0.005$, $\alpha = 1$, and initial coverage strength $\beta_0 = 1$. The coverage strength is linearly annealed over K_{RL} iterations:

$$\beta_k = \beta_0 \max\left(0, 1 - \frac{k}{K_{\text{RL}}}\right). \tag{23}$$

For policy optimization, coverage-induced sampling shift is corrected using the fixed coverage weight and standard PPO ratio in Eq. equation 11; only the PPO ratio is used for clipping. The policy is regularized toward the frozen reference model with $\lambda_{\text{KL}} = 0.01$. Tree construction and coverage correction are disabled at inference time.

Table 16: Main implementation and training hyperparameters of LongRx.

Configuration	Setting
Agent environment	
Maximum interaction steps $T_{\max}$	8
Candidates per retrieval K_{ret}	16
Warm-start	
SFT epochs	3
SFT learning rate	5×10^{-6}
SFT batch size	2
Gradient accumulation	16
SFT LoRA rank / alpha	16 / 32
CATO	
RL iterations	600
Cases per iteration	8
Maximum terminal leaves G	8
Branching factor B	3
Sampling temperature	0.9
Maximum generated tokens	192
RL learning rate	1×10^{-6}
PPO clipping range	0.2
Coverage smoothing α	1.0
Initial coverage strength β_0	1.0
Tree-credit coefficient λ_{tree}	0.5
Process coefficient λ_p	0.5
Interaction-cost coefficient λ_c	0.005
KL coefficient λ_{KL}	0.01
RL LoRA rank / alpha	8 / 16
LoRA dropout	0.05
Random seed	42

F.4 Optimization Settings

At inference time, tree construction, reward computation, and coverage correction are all disabled. LongRx directly follows the learned policy and interacts with the same frozen retrieval and verification tools until an `EXIT` request passes the terminal gate or the safeguard horizon $T_{\max}$ is reached.

G Additional Analysis of CATO

We provide additional analyses of Coverage-Aware Tree Optimization (CATO) to better understand the roles of opportunity-aware exploration and tree-relative credit assignment. We focus on three questions: (1) whether opportunity-aware coverage provides more effective exploration than standard or raw-count-based sampling, (2) whether tree-relative credit converts broader exploration into improved policy optimization, and (3) how the branching factor affects the performance–cost tradeoff.

G.1 Coverage and Tree-Credit Decomposition

Table 17 decomposes the main components of CATO under the same training and generation budget. **Raw Count Branching** increases the sampling probability of infrequent actions using only their selection counts. **Opportunity Coverage** additionally normalizes action usage by the number of sibling draws in which each action is valid. **Tree Credit Only** retains tree-relative sibling comparison without opportunity-aware branching.

Raw-count branching already improves exploration over Vanilla GRPO, but opportunity normalization provides a substantially larger increase in action coverage. This indicates that low visitation

Table 17: Detailed analysis of CATO on MIMIC-III. Raw Count rebalances actions using selection frequency alone, while Opportunity Coverage accounts for how often each action is available. All variants use the same training and generation budget.

Variant	Jaccard ↑	F1 ↑	DDI ↓	Train Cov. ↑	Steps ↓
Vanilla GRPO	27.16±0.28	40.73±0.22	10.22±0.08	0.48±0.02	3.11±0.05
+ Entropy Regularization	27.91±0.22	41.68±0.18	10.08±0.07	0.77±0.02	3.19±0.04
+ Temperature Sampling	28.02±0.21	41.81±0.17	10.05±0.06	0.80±0.01	3.21±0.04
+ Raw Count Branching	27.72±0.23	41.38±0.19	10.10±0.07	0.69±0.02	3.24±0.05
+ Opportunity Coverage	28.34±0.20	42.13±0.16	9.96±0.06	0.80±0.01	3.23±0.04
+ Tree Credit Only	28.17±0.21	41.94±0.18	9.99±0.06	0.63±0.02	3.18±0.04
+ Tree Credit + Entropy	28.44±0.20	42.30±0.17	9.96±0.06	0.78±0.02	3.19±0.04
+ Tree Credit + Temperature	28.58±0.18	42.43±0.15	9.93±0.05	0.81±0.01	3.20±0.03
Full CATO	**28.96±0.38**	**42.85±0.15**	**9.87±0.05**	**0.82±0.01**	3.20±0.03

Weight	Mean	Median	Std.	P95	Max	ESS/N	Clipped
w^{cov}	1.00	0.91	0.39	1.72	3.86	0.87	0.2%

Table 18: Distribution of local coverage weights during CATO training on MIMIC-III.

alone is insufficient to identify under-explored actions, since some actions may simply be unavailable in many states. In contrast, tree-relative credit yields a clear improvement in prescription quality even without achieving the same coverage as opportunity-aware branching. These results suggest that the two CATO components play complementary roles: coverage-aware branching broadens the decisions exposed during training, whereas tree-relative comparison improves learning from those alternatives. Combining both components achieves the strongest overall performance. Entropy regularization and higher-temperature sampling increase coverage but remain below Full CATO. More importantly, when each generic exploration method is combined with the same tree-relative credit, Full CATO remains 0.42 F1 points above the strongest alternative (Tree Credit + Temperature; paired 95% CI $[0.19, 0.65]$, Holm-adjusted $p = 0.004$). This controlled comparison isolates the incremental value of opportunity-normalized branching rather than attributing the gain to exploration or shared-state credit alone.

G.2 Coverage-Weight Diagnostics

The local coverage weights are normalized to unit mean. We report their dispersion and the normalized effective sample size $\mathrm{ESS}/N = (\sum_i w_i)^2/(N \sum_i w_i^2)$.

These diagnostics indicate that the local weights retain most of the effective sample size. They do not constitute correction for the full tree-induced state distribution.

G.3 Shared-Prefix Accounting and Sensitivity

CATO constructs the policy loss over unique sampled tree edges rather than flattening leaves into independent trajectories. Let $\mathcal{E}_g$ be the unique action edges in tree g. The tree contribution is

$$\mathcal{L}_{\mathrm{tree}} = \frac{1}{\sum_g |\mathcal{E}_g|} \sum_g \sum_{e \in \mathcal{E}_g} \ell_e, \tag{24}$$

so a shared prefix edge enters once. Its downstream value is the mean, rather than the sum, of descendant terminal rewards; a branch with more realized leaves therefore does not automatically receive greater loss weight. Concretely, for edge e with descendant leaves $\mathcal{D}(e)$, we define

$$A_e^{\mathrm{out}} = \frac{1}{|\mathcal{D}(e)|} \sum_{\ell \in \mathcal{D}(e)} A_\ell^{\mathrm{out}}, \qquad A_e = A_e^{\mathrm{out}} + \lambda_{\mathrm{tree}} A_e^{\mathrm{tree}} + \lambda_p \widehat{r}_e^{\mathrm{proc}} - \lambda_c c(u_e). \tag{25}$$

Loss accounting	F1 ↑	DDI ↓
Duplicate prefix per terminal leaf	42.31±0.20	10.02±0.07
Unique edge, descendant-value sum	42.57±0.18	9.95±0.06
Unique edge, descendant-value mean	**42.85±0.15**	**9.87±0.05**

Table 19: Sensitivity to shared-prefix and unequal-descendant accounting on MIMIC-III.

Leaf budget G	F1 ↑	DDI ↓	Relative generation cost
4	42.31±0.20	10.01±0.07	0.62×
6	42.64±0.17	9.94±0.06	0.81×
8	**42.85±0.15**	**9.87±0.05**	1.00×
12	42.89±0.18	9.86±0.06	1.47×

Table 20: Sensitivity to the terminal-leaf budget. $G = 8$ is the default.

The edge loss ℓ_e is the clipped PPO surrogate summed over the tokens of the complete parameterized action $u_e = (a_e, \xi_e)$ and scaled by its local coverage weight. Before entering the loss, coverage weights are clipped and renormalized to unit mean over the unique edges of each tree. Consequently, neither a long shared prefix nor an edge with more descendants receives extra weight merely because it is duplicated across terminal trajectories.

The differences between $G = 8$ and $G = 12$, and between weight caps 4 and 8, are within seed-level variation. We select $G = 8$ for lower generation cost and the cap of 4 for the higher effective sample size, rather than claiming either setting is uniquely optimal.

G.4 Objective and Schedule Sensitivity

We vary one coefficient at a time while retaining the default configuration for all other terms. The coverage schedule is also compared with non-vanishing endpoints to test whether annealing β to zero is responsible for the final result.

Performance is stable across the tested neighborhood. A schedule ending at $\beta = 0.2$ matches the default within uncertainty, showing that the result does not depend on CATO degenerating to ordinary tree GRPO at the end of training. We retain linear annealing to zero because it removes the exploratory sampling bias before convergence and has the lowest measured DDI proxy.

G.5 Exploration and Optimization Dynamics

Figure 6 further analyzes the training dynamics and branching behavior of CATO.

Action exploration. Figure 6(a) tracks normalized action coverage during RL training. Although Vanilla GRPO improves the policy objective, its action coverage gradually decreases, indicating increasingly concentrated exploration. Raw-count correction partially alleviates this behavior, while opportunity-aware coverage produces substantially broader exploration by distinguishing actions that are genuinely under-selected from those that are simply unavailable. Full CATO maintains the highest coverage throughout later stages of training.

Learning efficiency. Figure 6(b) compares terminal utility across optimization variants. Coverage-aware exploration alone consistently outperforms Vanilla GRPO by exposing the policy to a broader range of decision patterns. Tree-relative credit provides a further improvement by directly comparing alternative decisions originating from shared prescription states. Their combination yields the highest terminal utility, showing that broader exploration is most effective when accompanied by localized relative credit.

Branching factor. Figure 6(c) studies the number of sibling branches expanded from each state. Increasing the branching factor from $B = 1$ to $B = 3$ substantially improves F1 by exposing additional alternative decisions. Beyond $B = 3$, however, prescription quality largely saturates

Coverage-weight cap	F1 ↑	ESS/N ↑
2	42.66±0.17	0.94
4 (default)	**42.85±0.15**	0.87
8	42.84±0.18	0.79
No clipping	42.79±0.22	0.72

Table 21: Sensitivity to clipping of the local coverage weight.

Hyperparameter	Setting	F1 ↑	DDI ↓	Steps ↓
λ_{tree}	0.25 / **0.50** / 0.75	42.57±0.17 / **42.85±0.15** / 42.78±0.18	9.96 / **9.87** / 9.90	3.19 / 3.20 / 3.22
λ_p	0.25 / **0.50** / 0.75	42.53±0.18 / **42.85±0.15** / 42.70±0.19	9.98 / **9.87** / 9.89	3.20 / 3.20 / 3.21
λ_c	0.0025 / **0.005** / 0.010	42.87±0.17 / **42.85±0.15** / 42.61±0.20	9.86 / **9.87** / 9.94	3.39 / 3.20 / 2.89
α	0.5 / **1.0** / 2.0	42.79±0.18 / **42.85±0.15** / 42.76±0.19	9.89 / **9.87** / 9.91	3.21 / 3.20 / 3.20
β schedule	constant 0.5 / constant 1.0 / linear $\to 0.2$ / **linear** $\to 0$	42.55±0.19 / 42.60±0.18 / 42.82±0.16 / **42.85±0.15**	9.95 / 9.93 / 9.88 / **9.87**	3.20 / 3.22 / 3.21 / 3.20

Table 22: One-factor sensitivity analysis on MIMIC-III. Bold entries indicate the default configuration, not a statistically unique optimum.

while generation cost continues to increase. We therefore use $B = 3$ as the default setting, which provides a favorable balance between decision diversity and computational overhead.

Overall, these results support the two-stage design of CATO: opportunity-aware coverage determines which valid alternatives should receive additional exploration, while tree-relative outcomes determine which explored decisions should be reinforced. The improvement of Full CATO therefore cannot be attributed solely to broader action usage or a larger rollout tree, but instead arises from coupling structured exploration with shared-state credit assignment.

G.6 Backbone and Task Transfer

We finally test whether CATO is tied to a single 8B medication model. The backbone study keeps the environment, reward, and rollout budget fixed. The cross-task study applies only opportunity-aware branching and tree-relative credit to two public interactive benchmarks; medication-specific process and DDI rewards are disabled.

The consistent gains across backbone sizes and non-medication environments support CATO as a reusable exploration-and-credit mechanism. They do not imply universal improvement; environments without discrete valid-action sets or shared-state branching may not benefit from the same construction.

H Additional Method Details

This section provides additional details of the LongRx agent, reward design, and Coverage-Aware Tree Optimization (CATO). Implementation hyperparameters are summarized in Appendix F.

H.1 Agent State and Actions

At step t, the agent observes

$$s_t = (X, M_t, C_t, V_t, H_t), \tag{26}$$

where X is the patient EHR, M_t the current prescription, C_t the visible candidate set, V_t the latest valid verification feedback, and H_t the interaction history. Each episode starts from $M_0 = C_0 = V_0 = H_0 = \varnothing$.

The agent outputs a parameterized action $u_t = (a_t, \xi_t)$, where

$$a_t \in \{\texttt{RETRIEVE}, \texttt{ADD}, \texttt{DELETE}, \texttt{VERIFY}, \texttt{EXIT}\}. \tag{27}$$

For `RETRIEVE`, ξ_t is a generated query; for `ADD` and `DELETE`, it is a medication subset; `VERIFY` and `EXIT` require no argument. Actions that are not executable under the current state are masked.

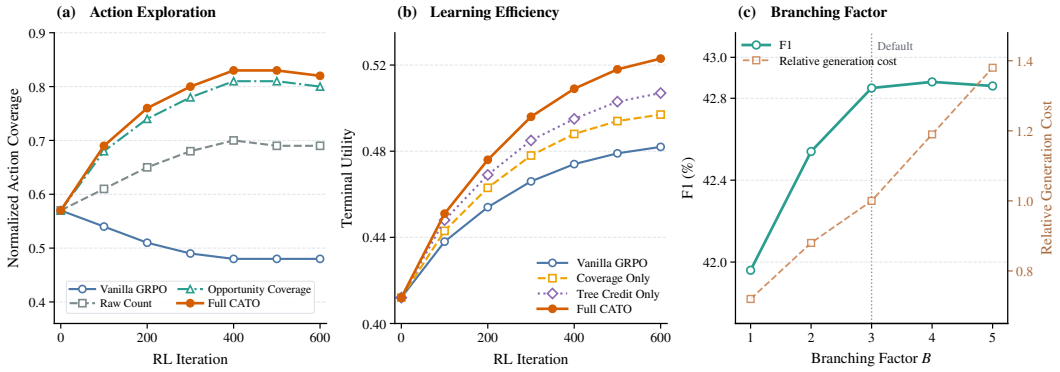

Figure 6: Additional analysis of CATO on MIMIC-III. **(a)** Opportunity-aware branching mitigates action exploration collapse and maintains substantially broader action coverage than Vanilla GRPO. **(b)** Coverage-aware exploration and tree-relative credit provide complementary optimization gains. **(c)** A moderate branching factor achieves a favorable performance–cost trade-off, while further increasing the number of branches yields diminishing performance gains.

Backbone	Tree GRPO F1	CATO F1
Qwen3-4B	39.84±0.24	**40.55**±0.20
Qwen3-8B	41.10±0.20	**42.01**±0.17
Aloe-Beta-8B	40.73±0.22	**42.85**±0.15

Table 23: Backbone transfer on MIMIC-III under matched rollout budgets.

H.2 State-Aware Retrieval and Verification

When `RETRIEVE` is selected, LongRx generates a state-conditioned query q_t and updates the candidate set using a frozen retrieval backend:

$$C_{t+1} = C_t \cup \big(\mathcal{R}(q_t) \setminus C_t\big). \tag{28}$$

Later retrievals may condition on the evolving prescription and previous tool feedback, allowing retrieval to remain part of the decision loop rather than a one-time preprocessing step.

`VERIFY` invokes a fixed DDI checker on the complete current prescription,

$$V_{t+1} = \mathcal{V}(M_t). \tag{29}$$

Any subsequent medication edit invalidates the previous verification result. An `EXIT` request terminates the episode only after the deterministic terminal check accepts the current prescription; otherwise the returned pairs are added to the state and interaction resumes.

H.3 Trajectory Warm-Start and Reward

LongRx is warm-started from outcome-filtered multi-step demonstrations generated by a stronger teacher using the same state, action, and tool interfaces. The teacher never observes the reference prescription Y during generation; trajectories are scored only after termination.

The terminal prescription reward is

$$R^{\text{final}} = w_f\,\text{F1}(M_T, Y) - w_d\,\text{DDI}(M_T) - w_m E_{\text{med}}(M_T, Y), \tag{30}$$

where E_{med} penalizes large prescription-size deviation.

For prescription-changing actions, we additionally use process feedback:

$$r_t^{\text{proc}} = \lambda_{\text{cov}}\Delta\,\text{Recall}_t + \lambda_{\text{red}}\Delta\,\text{Extra}_t + \lambda_{\text{safe}}\Delta\,\text{DDI}_t\,. \tag{31}$$

For `RETRIEVE`, `VERIFY`, and `EXIT`, $r_t^{\text{proc}} = 0$.

Benchmark	Metric	GRPO	Tree GRPO	CATO
HotpotQA tool agent	EM	48.6±0.4	50.1±0.3	**51.0**±**0.3**
WebShop	Success	42.8±0.6	44.1±0.5	**45.0**±**0.4**

Table 24: Task transfer of the generic CATO components (%). No medication-specific reward or action is used in either benchmark.

H.4 Coverage-Aware Tree Optimization

For each patient-level rollout tree g, CATO maintains $N_g(a)$, the number of times action a has been selected, and $O_g(a)$, the number of sibling draws in which a was valid. At state s_t, high-level actions are sampled according to

$$q_g(a \mid s_t) \propto \mathbf{1}[a \in \mathcal{A}_{\text{valid}}(s_t)]\pi_{\text{old}}(a \mid s_t) \left(\frac{O_g(a) + \alpha}{N_g(a) + \alpha} \right)^{\beta}. \tag{32}$$

CATO samples multiple sibling actions successively from this distribution, recomputing coverage counts after each draw. Action types may repeat, while their concrete arguments remain generated by the original conditional policy.

For a branch u_t from state s_t, let $\mathcal{L}(s_t, u_t)$ denote its descendant terminal leaves. Its downstream value is

$$Q_{\text{tree}}(s_t, u_t) = \frac{1}{|\mathcal{L}(s_t, u_t)|} \sum_{\ell \in \mathcal{L}(s_t, u_t)} R_\ell^{\text{final}}, \tag{33}$$

and the sibling-relative advantage is

$$A_{\text{tree}}(s_t, u_t) = \frac{Q_{\text{tree}}(s_t, u_t) - \mu_{\text{sib}}(s_t)}{\sigma_{\text{sib}}(s_t) + \epsilon}. \tag{34}$$

The final step-level advantage combines trajectory-, tree-, and process-level credit:

$$A_{i,t} = A_i^{\text{out}} + \lambda_{\text{tree}} A_{i,t}^{\text{tree}} + \lambda_p \widehat{r}_{i,t}^{\text{proc}} - \lambda_c c(u_{i,t}). \tag{35}$$

Coverage affects rollout exploration only and never enters the prescription reward. Because high-level actions are sampled from the coverage-adjusted behavior distribution, policy optimization uses the corresponding importance ratio and the standard clipped objective with KL regularization.

I Theoretical Analysis of CATO

We analyze the two components of Coverage-Aware Tree Optimization (CATO): opportunity-aware action exploration and tree-relative credit assignment. The coverage analysis concerns the high-level branching distribution, since concrete action arguments remain conditionally generated by the policy. The fixed-state equilibrium analysis below concerns a single repeated draw with fixed old-policy probabilities; each sibling draw in the implemented algorithm uses the same correction with updated coverage counts.

I.1 Opportunity-Aware Coverage Correction

Let

$$p_a = \pi_{\text{old}}(a \mid s) \tag{36}$$

denote the old-policy probability of a valid action a, and define its opportunity-normalized selection rate as

$$r_a = \frac{N_a + \alpha}{O_a + \alpha}. \tag{37}$$

The CATO branching distribution can be written as

$$q(a \mid s) \propto p_a r_a^{-\beta}. \tag{38}$$

Proposition 1 (Pairwise coverage correction). For any two valid actions a and b,

$$\frac{q(a \mid s)}{q(b \mid s)} = \frac{p_a}{p_b}\left(\frac{r_b}{r_a}\right)^{\beta} = \frac{p_a}{p_b}\left[\frac{O_a+\alpha}{N_a+\alpha}\frac{N_b+\alpha}{O_b+\alpha}\right]^{\beta}. \tag{39}$$

Proof. Substituting Eq. equation 38 into the ratio cancels the normalization constant, directly yielding Eq. equation 39. □

Hence, if $r_a > r_b$, action a has been selected more frequently relative to its opportunities, and for $\beta > 0$,

$$\frac{q(a \mid s)}{q(b \mid s)} < \frac{p_a}{p_b}. \tag{40}$$

CATO therefore suppresses actions that are over-represented relative to their availability rather than simply balancing raw visitation counts.

I.2 Fixed-State Exploration Behavior

We next consider an idealized fixed state with K valid actions and fixed old-policy probabilities $p_a > 0$. Since every action is valid at every step, all actions have equal opportunity counts, and the coverage rule reduces asymptotically to

$$q_a \propto p_a x_a^{-\beta}, \tag{41}$$

where x_a denotes the empirical selection frequency.

Theorem 1 (Coverage equilibrium). The unique interior fixed point is

$$\boxed{x_a^\star = \frac{p_a^{\frac{1}{1+\beta}}}{\sum_j p_j^{\frac{1}{1+\beta}}}.} \tag{42}$$

Proof. At equilibrium,

$$x_a^\star = \frac{p_a (x_a^\star)^{-\beta}}{Z}, \tag{43}$$

which implies

$$(x_a^\star)^{1+\beta} = \frac{p_a}{Z}. \tag{44}$$

Therefore $x_a^\star \propto p_a^{1/(1+\beta)}$; normalization gives Eq. equation 42. □

Corollary 1 (Log-odds contraction). For any two actions a and b,

$$\log \frac{x_a^\star}{x_b^\star} = \frac{1}{1+\beta} \log \frac{p_a}{p_b}. \tag{45}$$

Thus, $\beta > 0$ contracts pairwise action log-odds by $1/(1+\beta)$. When $\beta = 0$, the original policy distribution is recovered; increasing β progressively flattens the induced exploration distribution. Under this fixed-state idealization, the correction is therefore equivalent to a higher-temperature transformation of the old-policy action distribution.

I.3 Tree-Relative Credit

Coverage determines which alternative decisions are explored, while the tree structure determines how they are compared. For a shared state s, let $\mathcal{B}(s)$ denote its realized sibling branches and let

$$A_{\text{tree}}(s,u) = \frac{Q_{\text{tree}}(s,u) - \mu_{\text{sib}}(s)}{\sigma_{\text{sib}}(s) + \epsilon} \tag{46}$$

be the tree-relative advantage.

Proposition 2 (Centered sibling credit). For a sibling set with $|\mathcal{B}(s)| > 1$,

$$\sum_{u \in \mathcal{B}(s)} A_{\text{tree}}(s, u) = 0. \tag{47}$$

Proof. Because $\mu_{\text{sib}}(s)$ is the mean sibling value,

$$\sum_{u \in \mathcal{B}(s)} (Q_{\text{tree}}(s, u) - \mu_{\text{sib}}(s)) = 0. \tag{48}$$

Dividing by the common positive normalization term preserves the equality. □

Therefore, tree-relative credit does not reward all branches simply because their absolute terminal rewards are high. Instead, it identifies which decision performs better or worse than alternative decisions originating from the same prescription state.

Moreover, for any positive affine transformation

$$Q'_{\text{tree}} = cQ_{\text{tree}} + d, \qquad c > 0, \tag{49}$$

the sibling ordering and standardized tree advantages are unchanged up to the numerical stabilizer ϵ. This makes tree-relative credit primarily sensitive to local differences among alternative decisions rather than absolute reward scale.

I.4 Support and Importance Correction

For any valid action a, $\alpha > 0$, and finite β,

$$q_g(a \mid s) > 0 \quad \Longleftrightarrow \quad \pi_{\text{old}}(a \mid s) > 0, \tag{50}$$

because the coverage factor is strictly positive. Thus, CATO only reweights action types already supported by the old policy, while invalid actions are removed by the feasibility mask.

For sibling draw j, the rollout behavior distribution is

$$\mu^{(j)}_{\text{old}}(u \mid s) = q^{(j)}_g(a \mid s)\pi_{\text{old}}(\xi \mid s, a). \tag{51}$$

We separate the resulting sampling correction from the PPO update. Since the structured argument is sampled from the same conditional old policy, the coverage importance weight reduces to

$$w^{(j)}_{\text{cov}} = \frac{\pi_{\text{old}}(a \mid s)}{q^{(j)}_g(a \mid s)}. \tag{52}$$

The PPO ratio remains

$$r_t(\theta) = \frac{\pi_\theta(u_t \mid s_t)}{\pi_{\text{old}}(u_t \mid s_t)}. \tag{53}$$

CATO therefore weights the standard clipped PPO surrogate by $w^{(j)}_{\text{cov}}$ rather than clipping the behavior-corrected ratio directly. In particular, $r_t(\theta_{\text{old}}) = 1$, so coverage-aware sampling does not cause samples to enter the PPO clipping region before any policy update. This ratio only reweights the sampled high-level action conditional on a visited state. Tree expansion also changes which descendant states are visited, how many leaves descend from each branch, and the dependence among siblings; these effects are not corrected by $w^{(j)}_{\text{cov}}$. CATO should therefore be viewed as a practical tree-structured objective rather than an unbiased estimator of the original on-policy objective.

I.5 Discussion

The analysis characterizes the complementary roles of the two CATO components. Opportunity-aware branching applies negative feedback to actions that are over-selected relative to their availability and, under an idealized fixed-state setting, contracts action log-odds by $1/(1 + \beta)$. Tree-relative credit then converts the resulting sibling branches into centered, state-conditioned supervision by comparing downstream outcomes from the same prescription state.

These results characterize CATO's exploration and credit-assignment mechanisms, not prescription quality itself. CATO does not assume that rarely selected actions are beneficial or that uniform action usage is desirable. Coverage only determines which alternatives receive sufficient exploration; their downstream prescription outcomes determine whether they are ultimately reinforced.

Metric	Difference (pp)	95% paired bootstrap CI	p-value
F1	$+1.06$	$[0.82, 1.30]$	< 0.001
DDI	-0.44	$[-0.61, -0.27]$	< 0.001

Table 25: Patient-level paired comparison between LongRx and the strongest baseline on the MIMIC-III test set. Metrics are averaged across three seeds for each patient before paired statistical testing.

Target	Metric	Difference (pp)	95% paired bootstrap CI	Holm p
MIMIC-IV	Jaccard	$+1.08$	$[0.71, 1.45]$	< 0.001
MIMIC-IV	F1	$+1.27$	$[0.91, 1.63]$	< 0.001
MIMIC-IV	DDI	-0.71	$[-0.95, -0.47]$	< 0.001
eICU	Jaccard	$+0.83$	$[0.49, 1.17]$	< 0.001
eICU	F1	$+0.88$	$[0.52, 1.24]$	< 0.001
eICU	DDI	-0.59	$[-0.82, -0.36]$	< 0.001

Table 26: Patient-level paired transfer comparison between LongRx and RES-MR. Intervals are computed after averaging each patient's prediction over seeds.

J Additional Analyses and Statistical Testing

Patient-level uncertainty and statistical testing. All paired analyses use exactly the same held-out MIMIC-III test patients and seed-level predictions as Table 1; no additional test-set subsampling is performed. We evaluate three training seeds (13, 21, and 42). For each patient, we first average the metric across seeds and then perform a two-sided paired t-test and a paired bootstrap confidence interval with 10,000 patient resamples. LongRx improves F1 over the strongest baseline by 1.06 percentage points (95% CI $[0.82, 1.30]$; paired $t(797) = 8.68$, $p < 0.001$) and reduces DDI rate by 0.44 percentage points (95% CI $[-0.61, -0.27]$; paired $t(797) = -5.08$, $p < 0.001$).

Cross-dataset paired uncertainty. We apply the same patient-level paired bootstrap procedure to the transfer predictions and report Holm-adjusted intervals for the three metrics within each target dataset.

Prescription-size and subgroup analysis. We further examine whether the improvement in prescription overlap is accompanied by systematic changes in prescription size and over-recommendation. Table 27 reports diagnostics computed from the same predictions as Table 1. LongRx slightly reduces the predicted prescription size and absolute size error while improving both precision and recall. We additionally report the number of medications absent from the reference prescription and the absolute number of DrugBank-listed interacting pairs.

Controlled operating points. To separate ranking quality from prescription length, operating points are selected on the validation split and then frozen for the test set. RES-MR uses its medication probabilities. For LongRx, each proposed medication receives the length-normalized log probability of its most recent valid `ADD` argument; a single global threshold and cardinality penalty are tuned on the validation split, after which the final set is deterministically replayed and reverified without test labels. We compare methods at the reference-set size and at matched precision.

Blinded review of additional recommendations. Two clinical reviewers independently assess 250 medications per method that are recommended but absent from the recorded prescription. They are blinded to the method and classify each recommendation using only pre-decision EHR data; disagreements are adjudicated by a third reviewer.

The controlled operating points preserve the quality and DDI advantages when prescription length or precision is matched. The adjudication study further distinguishes absence from the historical record from lack of clinical support; it remains a retrospective judgment rather than an outcome evaluation.

Method	Ref. #Med	Pred. #Med	Precision	Recall	Abs. size error	#Extra	#DDI pairs	F1
Strongest baseline	14.78	22.71	35.0	51.9	9.28	14.76	24.9	41.79
LongRx	14.78	22.54	35.4	54.2	9.04	14.56	23.6	42.85

Table 27: Prescription diagnostics on the MIMIC-III test set. Precision, recall, and F1 are percentages. #Extra denotes the mean number of predicted medications absent from the recorded prescription, and #DDI pairs denotes the mean number of DrugBank-listed interacting pairs in the final prescription.

Setting	Method	Jaccard ↑	F1 ↑	Precision ↑	Recall ↑	DDI ↓	#Med
Default	RES-MR	28.03	41.79	35.0	51.9	10.31	22.71
Default	LongRx	**28.96**	**42.85**	**35.4**	**54.2**	**9.87**	22.54
Size-matched (target $\#\text{Med} = 14.78$)	RES-MR	26.14	39.62	39.9	39.4	9.72	14.81
Size-matched (target $\#\text{Med} = 14.78$)	LongRx	**27.33**	**40.88**	**41.2**	**40.6**	**9.11**	14.82
Precision-matched (35.0%)	RES-MR	28.03	41.79	35.0	51.9	10.31	22.71
Precision-matched (35.0%)	LongRx	**28.73**	**42.62**	35.0	**54.5**	**9.79**	22.88

Table 28: MIMIC-III results under matched prescription size and precision. Operating points are selected using validation data only.

Risk-stratified diagnostics. The aggregate DrugBank DDI rate does not account for dose, route, monitoring, or clinical outcomes. We therefore additionally stratify listed interactions by severity and report patient-level allergy and contraindication conflicts. Interaction severity is assigned by a frozen DrugBank severity crosswalk; unmapped pairs remain in the "Unknown" category. Allergy strings are normalized to generic ingredients before exact ingredient or ingredient-family matching. Contraindications are evaluated with a preregistered medication–ICD rule set derived only from training-independent drug documentation.

The severity, allergy, and contraindication mappings cover 95.1%, 93.8%, and 98.4% of eligible test records, respectively; unmapped records are excluded only from the corresponding patient-context denominator.

The DDI reduction is concentrated in major interactions, while minor rates remain similar. These retrospective proxies narrow the interpretation of safety but do not establish clinical suitability.

Table 32 further stratifies patients by reference prescription size, diagnosis frequency, and the presence of a known DDI pair in the reference prescription. The improvement is consistent across subgroups, although the gain is smaller for patients with larger reference prescriptions.

Terminal-verification audit. An agent-requested `VERIFY` remains optional and is counted in the action statistics. Separately, each `EXIT` request invokes a deterministic terminal gate outside the learned action budget. The gate checks the current prescription; if it returns a listed interaction not acknowledged in the current verification state, the feedback is shown to the policy and the episode resumes. Any subsequent edit invalidates the result, so an accepted `EXIT` always has a fresh final check. At the safeguard horizon, the same check is recorded immediately before returning the current set; this guarantees fresh safety metadata but does not claim that every listed interaction has been removed.

Verification-feedback control. We isolate the effect of observing DDI verification feedback by comparing two policies trained with identical initialization, data, DDI reward, rollout budget, and random seeds; the control masks only the verification result shown to the policy. With feedback enabled, 38.6% of verification calls are followed by a medication edit, compared with 17.2% when feedback is masked. Among verification-followed edits, the mean number of interacting pairs decreases by 0.38, compared with 0.11 for the masked control. The feedback-enabled policy also achieves a lower overall DDI rate (9.87% vs. 10.42%; paired difference -0.55 percentage points, 95% CI $[-0.73, -0.37]$, $p < 0.001$).

Method	Supported	Plausible	Unsupported	High-risk
RES-MR	147 (58.8)	31 (12.4)	65 (26.0)	7 (2.8)
LongRx	**172 (68.8)**	33 (13.2)	**42 (16.8)**	**3 (1.2)**

Table 29: Blinded review of additional recommendations, reported as count (percentage) among 250 mutually exclusive final judgments per method. Inter-reviewer agreement before adjudication is Cohen's $\kappa = 0.76$. The 95% Wilson intervals for the supported category are $[52.6, 64.7]$ for RES-MR and $[62.8, 74.2]$ for LongRx.

Method	Major DDI	Moderate DDI	Minor DDI	Unknown	Allergy conflict	Contraindication
RES-MR	2.05	6.72	1.01	0.53	1.82	1.24
LongRx	**1.71**	**6.59**	1.05	**0.52**	**1.39**	**1.05**

Table 30: Risk-stratified diagnostics on MIMIC-III. DDI columns are pair-level percentages and sum to the aggregate DDI rate; the final two columns are patient-level percentages.

We further cross the DDI reward with visibility of verification feedback to separate their contributions.

Importantly, this comparison is distinct from the LongRx–RES-MR comparison above: the -0.55 percentage-point difference compares LongRx with its feedback-masked control, whereas the corresponding LongRx–RES-MR difference in Table 1 is -0.44 percentage points.

Verifier-access and external-DDI controls. To separate policy quality from privileged access to the evaluation matrix, we give RES-MR the same DrugBank checker and a matched medication-deletion budget. We also evaluate every frozen output against a disjoint TWOSIDES-derived DDI matrix that is never exposed to training, reward computation, or inference.

The matched filter narrows the DrugBank gap but does not remove LongRx's F1 or external-DDI advantage. This result supports a narrower claim: closed-loop training improves the quality–interaction-proxy trade-off beyond merely querying the same matrix, while neither DDI source establishes clinical safety.

Inference cost against fixed workflows. We compare realized set-edit/tool cost, generated tokens, and wall-clock latency under the same retrieval backend, verifier, 8B policy, batch size one, and hardware. The deterministic terminal gate is included in tokens and latency and in tool cost for every method, but is not counted as a learned action step.

LongRx is not the least expensive configuration; instead, it lies on the quality–safety–cost frontier and improves over the longest fixed workflow at lower realized cost and latency.

Training and rollout cost. We additionally account for the computational cost of CATO under the same maximum terminal-trajectory budget as the matched GRPO baseline. Both methods generate at most eight terminal trajectories per patient. CATO shares prefixes across sibling continuations and therefore incurs a similar overall rollout-token budget despite explicitly branching from intermediate states. Table 38 reports the realized rollout statistics, while Table 39 summarizes total training cost.

K Prompt Templates

LongRx uses one shared prompt interface for teacher-trajectory generation, supervised fine-tuning, reinforcement-learning rollouts, and inference. In accordance with the closed-loop formulation in Section 4, the agent starts with an empty visible candidate set and an empty prescription; there is no separate drug-level filtering prompt or preconstructed initial prescription. At every step, the prompt consists of a fixed system instruction and a state-dependent observation. The reference prescription Y, missing or extra medications, and all reward values are excluded from both parts of the prompt.

Risk metric	Difference (pp)	95% paired CI
Major DDI	-0.34	$[-0.47, -0.21]$
Allergy conflict	-0.43	$[-0.61, -0.25]$
Contraindication	-0.19	$[-0.31, -0.07]$

Table 31: Paired uncertainty for risk-stratified diagnostics. Intervals use 10,000 patient bootstrap samples with Holm correction.

Patient subgroup	Baseline F1	LongRx F1	Difference (pp)
Reference size: ≤ 10	43.18	44.29	$+1.11$
Reference size: 11–20	41.96	43.12	$+1.16$
Reference size: > 20	39.12	39.87	$+0.75$
Rare diagnosis (bottom quartile)	35.71	36.88	$+1.17$
Reference contains DDI pair	40.84	41.91	$+1.07$
No reference DDI pair	42.36	43.42	$+1.06$

Table 32: Stratified F1 scores on MIMIC-III. Diagnosis-frequency cutoffs are computed from the training split, and DDI strata are defined by the recorded reference prescription.

K.1 System Instruction

The following system instruction defines the available tools, their constraints, and the exact output schemas. The same instruction is used at all decision steps.

LongRx system instruction

```
You are LongRx, a closed-loop medication recommendation agent.

Construct a clinically appropriate medication set using only the
    patient EHR,
the visible medication candidates, the current prescription, the
    interaction
history, and deterministic tool feedback. The reference prescription
    is hidden
and must never be requested or assumed.

At each step, choose exactly one action from `valid_actions`:
- RETRIEVE: generate a non-empty state-conditioned search query. The
    frozen
  retrieval backend returns at most 16 previously unseen medication
    IDs. Cost: 1.
- ADD: add one or more medication IDs. Every ID must be visible and
    absent from
  the current prescription. Cost: the number of medications added.
- DELETE: delete one or more medication IDs. Every ID must be in the
    current
  prescription. Cost: the number of medications deleted.
- VERIFY: check the complete current prescription for drug-drug
    interactions.
  The result remains valid only until the next successful ADD or
    DELETE. Cost: 1.
- EXIT: request termination. A deterministic terminal verification
    gate may
  return interaction feedback and resume the episode. Policy-action
    cost: 0.

Respect `remaining_steps`. Do not invent medication IDs, expose
    hidden labels,
```

Verification statistic	Rate
Episode with agent-requested `VERIFY`	48.7%
`EXIT` accepted on first terminal check	88.4%
Episode resumed after terminal feedback	11.6%
Post-verification edit followed by re-verification	100.0%
Final prescription with a fresh check	100.0%
Stale verification at accepted termination	0.0%

Table 33: Coverage and validity of verification at test time. Terminal-gate calls are deterministic environment checks and are reported separately from agent-requested actions.

Condition	DDI rate (%)	Verify→Edit (%)	Pair-count change
Feedback enabled	9.87	38.6	−0.38
Feedback masked	10.42	17.2	−0.11

Table 34: Controlled analysis of verification feedback. Pair-count change is computed only for verification calls followed by a prescription edit.

```
or output clinical reasoning. Return exactly one JSON object with no
    markdown,
explanation, or additional text.

Valid JSON schemas:
{"action":"RETRIEVE","query":"non-empty state-conditioned query"}
{"action":"ADD","medication_ids":[14,92]}
{"action":"DELETE","medication_ids":[130]}
{"action":"VERIFY"}
{"action":"EXIT"}
```

This instruction exposes the action cost to the policy but does not reveal the terminal or process reward. A single `ADD` or `DELETE` may contain multiple medication identifiers, consistent with the unordered set-valued output space.

K.2 State Observation

At step t, the user message serializes the observable state $s_t = (X, M_t, C_t, V_t, H_t)$ as a JSON object. Its fields are listed below.

The canonical user-message template is:

LongRx state observation template

```
CURRENT_STATE
{
  "ehr": {{patient_ehr}},
  "current_prescription": {{sorted_medication_entries_in_M_t}},
  "visible_candidates": {{sorted_medication_entries_in_C_t}},
  "verification": {{latest_valid_verification_or_null}},
  "history": {{previous_actions_and_tool_observations}},
  "remaining_steps": {{T_max_minus_t}},
  "valid_actions": {{state_dependent_valid_action_types}}
}
```

At the initial state, both `current_prescription` and `visible_candidates` are empty, `verification` is `null`, and `history` is empty. The model-specific chat wrapper is applied through the backbone tokenizer's native chat template; we do not hard-code model-specific tags

DDI reward	Verification feedback	F1 ↑	DDI ↓
No	Masked	41.78	10.89
Yes	Masked	42.31	10.42
No	Visible	42.04	10.28
Yes	Visible	**42.85**	**9.87**

Table 35: Factorial decomposition of DDI reward and verification feedback on MIMIC-III.

Condition	F1 ↑	DrugBank DDI ↓	External DDI ↓	#Med
RES-MR	41.79±0.16	10.31±0.06	8.74±0.08	22.71
RES-MR + matched verifier/filter	41.37±0.18	9.94±0.05	8.53±0.07	22.19
LongRx, verification feedback masked	42.31±0.17	10.42±0.07	8.48±0.06	22.60
LongRx	**42.85**±0.15	**9.87**±0.05	**8.19**±0.05	22.54

Table 36: Fair-access and external-matrix DDI controls on MIMIC-III. The matched filter receives the same checker and deletion budget as LongRx. External DDI is evaluated with a matrix unavailable to all policies.

such as `[INST]`. Medication metadata is obtained from the fixed medication vocabulary and is independent of the patient-specific reference prescription.

K.3 Action Output and Validity

The agent returns exactly one JSON object. Table 41 shows the accepted schemas and state-dependent constraints.

The environment rejects malformed JSON, unknown actions, additional fields, empty medication lists, and medication identifiers that violate the current state constraints. An invalid output leaves M_t, C_t, and V_t unchanged, but appends an invalid-action record to the history, consumes one decision step, and incurs unit cost.

K.4 Tool Feedback

Tool outputs are deterministic environment observations. A successful `RETRIEVE` returns only newly exposed medication identifiers:

Retrieval feedback

```
{"retrieved_medication_ids":[14,31,92]}
```

A successful edit reports the identifiers actually added or deleted:

Editing feedback

```
{"added_medication_ids":[14,92]}
{"deleted_medication_ids":[130]}
```

For `VERIFY`, the environment checks all unordered medication pairs in the complete current prescription using the fixed DDI matrix and returns:

Verification feedback

```
{
  "ddi_rate": 0.067,
  "interacting_pairs": [[12,31]]
}
```

Workflow	Steps	Edit/tool cost	Tokens	Latency (s)	F1 ↑	DDI ↓
R-A	2.00	22.6	610	1.42	40.68	10.63
R-A-V	3.00	23.6	690	1.71	41.19	10.18
R-A-V-D	4.00	25.7	780	2.13	41.72	9.99
R-A-V-R-A	5.00	31.6	1,060	2.84	42.05	10.02
LongRx	3.20	26.4	760	1.96	**42.85**	**9.87**

Table 37: Inference cost and performance of dynamic and predefined workflows on MIMIC-III. Latency is measured per patient on the same A100 GPU.

Rollout statistic	LongRx	Matched GRPO
Terminal trajectories / patient	8.00	8.00
Expanded action nodes / patient	23.7	24.9
Generated output tokens / patient	1.18K	1.21K
Relative rollout-token cost	0.98×	1.00×

Table 38: Realized reinforcement-learning rollout budget on MIMIC-III. Both methods use the same maximum of eight terminal trajectories per patient; token counts include generated action and argument tokens.

This feedback contains neither drug relevance labels nor comparisons with Y. After any successful `ADD` or `DELETE`, the stored verification field is reset to `null`; a later safety judgment therefore requires verification of the updated complete prescription.

K.5 Use Across Training and Inference

The prompt interface is identical across all stages. During trajectory warm-start, the teacher receives only the system instruction and observable state above; complete trajectories are scored against Y only after termination. SFT decomposes retained trajectories into state–action pairs and masks every system and state-observation token, so the language-model loss is applied only to the JSON action and its argument. During CATO, the coverage mechanism changes only the sampling distribution over the five high-level action types; medication subsets and retrieval queries are still generated from the policy conditioned on the same prompt. At inference, the same prompt is used without labels, rewards, or coverage correction until an `EXIT` request passes the terminal gate or the policy reaches $T_{\max}$.

L Notation Summary

Table 42 summarizes the main notation used throughout the paper.

M Extended Related Work

M.1 Medication Recommendation

Medication recommendation aims to identify a clinically appropriate set of medications from patient-specific information, including current diagnoses, historical visits, previous prescriptions, laboratory findings, and safety factors. Existing studies have gradually evolved from visit-level multi-label prediction to longitudinal modeling, knowledge-enhanced recommendation, and large language model (LLM)-based generation Ali et al. (2023); Zhang et al. (2026b); Zhu et al. (2024).

Early methods mainly formulate medication recommendation as a multi-instance multi-label classification problem. LEAP models the dependencies among medication labels using information from the current visit Zhang et al. (2017). However, visit-level methods usually treat each encounter independently and have limited ability to capture treatment evolution across hospitalizations. Longitudinal methods address this limitation by incorporating historical diagnoses and prescriptions. GAMENet integrates electronic health record information with a drug–drug interaction (DDI) graph

Stage	LongRx	Matched-data baseline
Teacher trajectory generation	46.2	46.2
Trajectory scoring and selection	1.8	1.8
RL rollout generation	32.4	32.1
8B policy optimization	18.1	17.6
Total A100 GPU hours	98.5	97.7

Table 39: Approximate training cost in A100 GPU hours under matched teacher-data and terminal-trajectory budgets.

Table 40: Fields in the LongRx state observation.

Field	**Content**
`ehr`	Patient history and current-visit clinical context X.
`current_prescription`	Sorted entries in M_t, each containing a medication identifier and fixed catalog metadata such as its generic name.
`visible_candidates`	Sorted entries in C_t with the same fixed medication metadata.
`verification`	Latest valid complete-prescription DDI result V_t, or `null` if no valid result is available.
`history`	Previous actions, validity indicators, tool observations, and incurred action costs.
`remaining_steps`	Number of decisions remaining before $T_{\max}$.
`valid_actions`	Action types executable in the current state.

through a graph-augmented memory network Shang et al. (2019). COGNet further introduces a copy-or-predict mechanism to reuse historical prescriptions when making current medication decisions Wu et al. (2022), while MICRON explicitly models medication changes between consecutive visits using recurrent residual learning Yang et al. (2021a). These approaches demonstrate the importance of patient history, but still generally produce medication predictions within a fixed feed-forward pipeline.

Another line of work incorporates pharmacological knowledge and molecular structures to improve the safety and specificity of recommendation. SafeDrug employs dual molecular encoders to represent drug structures and uses them to constrain potentially unsafe combinations Yang et al. (2021b). MoleRec models the interaction between patient conditions and fine-grained molecular substructures through substructure-aware attention Yang et al. (2023). These methods improve drug-level representation and DDI modeling, but the recommendation process remains largely determined by predefined scoring or decoding mechanisms. They do not explicitly model how a decision-maker should inspect, revise, verify, or terminate a prescription.

Recent studies focus on improving patient representations and addressing the limitations of structured EHR codes. RAREMed adopts a Transformer-based representation learning framework to improve performance on rare diseases and underrepresented patient groups Zhao et al. (2024). NLA-MMR performs multimodal alignment between patient and medication representations, allowing information from different modalities to be jointly utilized Tan et al. (2024). These methods enrich the information available to the recommender, but their outputs are still commonly evaluated as a single medication list rather than as the result of an interactive decision process.

The emergence of LLMs provides a natural language interface for integrating structured records, clinical notes, and medical knowledge. LEADER explores knowledge distillation for LLM-based medication recommendation Liu et al. (2024), while LAMO combines structured diagnoses and procedures with unstructured clinical text to capture richer patient context and reduce overprescription Zhao et al. (2025). FLAME reformulates medication generation as a sequence of drug-level decisions and introduces step-wise policy optimization to provide more localized training signals Fan et al. (2026). SafeRx-Agent further emphasizes knowledge grounding and explicit safety verification through an agentic framework Wang et al. (2026b). Despite these advances, most existing methods either generate the medication list in a single pass or use a predefined drug-level decoding process. In particular, they do not explicitly represent the patient as an evolving environment in

Table 41: LongRx action schemas. No fields other than those shown are accepted.

Action	JSON output	Validity constraint
RETRIEVE	{"action":"RETRIEVE","query":"..."}	query must be a non-empty state-conditioned string.
ADD	{"action":"ADD","medication_ids":[...]}	The non-empty ID set must be a subset of $C_t \setminus M_t$.
DELETE	{"action":"DELETE","medication_ids":[...]}	The non-empty ID set must be a subset of M_t.
VERIFY	{"action":"VERIFY"}	The current prescription must be non-empty.
EXIT	{"action":"EXIT"}	Requests the deterministic terminal verification gate.

Table 42: Summary of the main notation.

Symbol	Description
$\mathcal{M}$	Medication vocabulary
X	Patient EHR available before the medication decision at the target encounter
Y	Recorded reference prescription
M_t	Current prescription at step t
C_t	Visible medication candidate set at step t
V_t	Latest valid DDI verification feedback
H_t	Interaction history
s_t	Agent state (X, M_t, C_t, V_t, H_t)
a_t	High-level action type at step t
ξ_t	Action-specific argument
u_t	Parameterized action (a_t, ξ_t)
$\mathcal{A}_{\text{valid}}(s_t)$	Valid high-level actions at state s_t
$\mathcal{R}$	Frozen medication retrieval backend
$\mathcal{V}$	Fixed DDI verification tool
R^{final}	Terminal prescription reward
r_t^{proc}	Process-level prescription feedback
$c(u_t)$	Interaction cost of action u_t
A_i^{out}	Trajectory-level group-relative advantage
$N_g(a)$	Number of times action a is selected as a branch
$O_g(a)$	Number of sibling draws where action a is valid
$q_g(a \mid s_t)$	Coverage-aware branching distribution
α	Coverage smoothing constant
β	Coverage correction strength
$\mathcal{L}(s_t, u_t)$	Terminal leaves descending from branch u_t
$Q_{\text{tree}}(s_t, u_t)$	Downstream value of a tree branch
$A_{\text{tree}}(s_t, u_t)$	Sibling-relative tree advantage
λ_{tree}	Weight of tree-relative credit

which the agent decides whether to acquire more information, edit the current prescription, perform a safety check, or stop. LongRx addresses this gap by formulating medication recommendation as a bounded interaction process between a single agent and a patient-conditioned prescription environment, with the final output treated as a medication set and the intermediate behavior optimized through editing, verification, and termination decisions.

M.2 LLM Agents and Long-Horizon Policy Optimization

LLM agents extend language models from static text generation to sequential decision making. An agent receives observations from an external environment, reasons about the current state, invokes tools or APIs, and updates its behavior based on the resulting feedback. This interaction introduces

challenges that do not arise in ordinary single-turn generation, including partial observability, accumulating state, invalid actions, delayed outcomes, and long-horizon credit assignment Du et al. (2026); Zhang et al. (2025); Chang et al. (2026).

Early agent frameworks established the basic reasoning–action loop. ReAct interleaves natural-language reasoning with environment actions, enabling a language model to use observations to revise its next decision Yao et al. (2022). Toolformer trains language models to determine when to invoke external tools, which tool to use, and how to incorporate the returned results Schick et al. (2023). WebGPT applies this paradigm to browser navigation and optimizes answer quality using feedback from the browsing trajectory Nakano et al. (2021). These works demonstrate that tool invocation and environmental feedback can be integrated into language-model generation, but their main objective is task completion rather than structured optimization of intermediate decisions.

A second line of research studies planning, reflection, and persistent experience. Reflexion stores textual feedback from previous attempts and uses it to improve subsequent behavior Shinn et al. (2023). Tree-of-Thoughts explores multiple reasoning branches before selecting a final solution, while Voyager combines automatic curriculum generation, a skill library, and environment feedback for open-ended embodied tasks. Memory-oriented agents further introduce external or hierarchical memory to maintain information beyond a single context window. These approaches improve long-term consistency, but they typically rely on handcrafted reflection or search procedures rather than learning a domain-specific interaction policy.

Recent work applies reinforcement learning to optimize LLM agents directly. Agent-R1, Search-R1, and ReTool learn interactive reasoning and tool-use policies from outcome-level environmental feedback Cheng et al. (2025); Jin et al. (2025); Feng et al. (2026). However, assigning the same terminal reward to every intermediate action provides limited information about which decision caused success or failure. To address this issue, iStar, IGPO, HiPRAG, and WebArbiter introduce implicit step rewards, information-gain signals, retrieval-quality supervision, or generative process evaluation Liu et al. (2026); Wang et al. (2025); Wu et al. (2026); Zhang et al. (2026a). These methods improve the granularity of feedback, but their process signals are usually defined for search, retrieval, or general tool-use tasks and are not specialized for set-valued clinical decisions.

Another direction focuses on exploration and optimization under long horizons. GiGPO and Tree-GRPO compare alternative actions that share the same intermediate state or prefix, reducing the variance of group-relative policy updates Ji et al. (2026). CATO adopts this shared-state comparison principle rather than claiming a new sibling-relative estimator; its distinction is to allocate sibling draws according to opportunities among currently valid action types. ATPO adaptively allocates exploration to uncertain states, while DAPO and the Optimal Token Baseline improve optimization stability and variance control Cao et al. (2026); Yu et al. (2026). Recent surveys further emphasize that practical agent training requires jointly considering reward design, trajectory construction, environment stability, and interaction cost. Nevertheless, most existing methods evaluate trajectories using scalar task success or generic process quality. They do not explicitly account for the fact that medication recommendation produces an unordered but globally coupled set, where adding or removing one medication changes the quality and safety of the entire prescription.

LongRx builds on this line of work while targeting the specific structure of medication recommendation. It models the patient record, current prescription, candidate medications, and verification results as an evolving environment state. A single agent can perform medication-set edits, request verification, and terminate the interaction within a bounded budget. The training objective combines final set quality, prescription safety, and interaction efficiency, while shared-state rollouts provide relative comparisons between alternative editing decisions. This formulation connects long-horizon agent optimization with the set-valued and safety-critical nature of clinical medication recommendation.

ETHICS STATEMENT

This retrospective study uses de-identified MIMIC-III, MIMIC-IV, and eICU records under their respective credentialing and data-use requirements. Patient-level inputs, generated trajectories, tool outputs, and derived artifacts may still contain sensitive clinical context and should therefore be

stored, accessed, and shared only under the applicable data-use restrictions. Any released examples should be reviewed to avoid disclosure of residual identifying information.

The evaluation metrics used in this work are retrospective proxies rather than direct measures of clinical benefit. Prescription overlap reflects agreement with recorded historical treatment, while the DrugBank-based DDI metric captures only listed pairwise interactions and does not account for dosage, route, interaction severity, indication, allergies, organ function, monitoring, or patient outcomes. Accordingly, neither higher prescription overlap nor lower DDI rate should be interpreted as evidence that a recommendation is clinically appropriate for an individual patient.

LongRx is developed solely as a research system and is not intended for autonomous prescribing or direct clinical decision making. Prospective clinical use would require independent clinical validation, evaluation across institutions and patient subgroups, clinician oversight, and appropriate institutional and ethical review.

LLM USAGE

During manuscript preparation, an LLM assistant was used to improve English wording and clarity, suggest organization, and construct explicitly marked synthetic stress-test examples for internal planning. Those placeholders are not empirical evidence and must be removed or replaced by author-run results before submission. The central research idea and LongRx method were developed by the authors; the authors remain responsible for verifying every submitted claim, computation, figure, and citation.