Extended Abstract

Motivation Clinical prediction from EHRs depends critically on how longitudinal patient data is transformed into model inputs. Naive serializations of raw EHR timelines bury task-relevant signals within heterogeneous, high-volume records. Demirel et al. Demirel et al. (2026) address this through rubric-based representation learning (LRRL), synthesizing task-specific structured extraction schemas that substantially outperform naive baselines on EHRSHOT Wornow et al. (2023). However, rubric induction remains one-shot with no mechanism for downstream feedback. We present REFINE (Reinforcement-based EHR Feature Induction and Editing), which applies GRPO to iteratively refine rubric schemas guided by an AUROC reward, without any additional LLM calls during training.

Method We frame rubric refinement as a Markov decision process where the state $s_t = (R_t, \mathcal{D}, \tau)$ captures the current rubric R_t, patient dataset $\mathcal{D}$, and task τ. Actions correspond to six deterministic rubric edits resolved by heuristics on the filled rubric matrix: `remove_lowest_variance_field`, `add_direction_of_change`, `add_missingness_indicator`, `refine_granularity`, `remove_redundant_pair`, and `split_highest_entropy_field`. At each step the selected action is applied to the current best rubric state, affected fields are re-filled for 400 stratified patients, representations are re-embedded using the LRRL pipeline, a logistic regression classifier is retrained, and AUROC is evaluated on a held-out validation set. We train a categorical policy over the six actions using GRPO, computing group-normalized advantages $A_i = (r_i - \bar{r})/\mathrm{std}(\mathbf{r})$ from exhaustive group rollouts at each step. A best-state rollout strategy branches the trajectory from the highest-AUROC rubric seen so far, preventing compounding degradation from suboptimal selections. Training stops when best-state AUROC improves by less than 0.01 for two consecutive steps, with a maximum horizon trajectory length of $T = 10$.

Implementation REFINE is implemented as a fork of the LRRL codebase, with each action implemented as a standalone Python script requiring no additional generative LLM calls during the training loop. Experiments are run on a single A100 40GB GPU across four EHRSHOT tasks: length-of-stay (`guo_los`), 30-day readmission (`guo_readmission`), hyperkalemia anticipation (`lab_hyperkalemia`), and new lupus diagnosis (`new_lupus`). The policy is optimized with learning rate $\eta = 0.3$, selected via sweep over $\{0.05, 0.1, 0.2, 0.3\}$, and each configuration is run with $n = 3$ random seeds with results reported as mean $\pm$ standard error.

Results REFINE improves AUROC over the one-shot LRRL baseline on all four tasks at $\eta = 0.3$: `guo_los` $0.653 \rightarrow 0.723$ $(+0.070)$, `guo_readmission` $0.713 \rightarrow 0.717$ $(+0.004)$, `lab_hyperkalemia` $0.699 \rightarrow 0.742$ $(+0.043)$, and `new_lupus` $0.592 \rightarrow 0.595$ $(+0.003)$. Higher learning rates consistently outperform lower rates, with $\eta = 0.3$ converging in 3–4 steps on the two strongest tasks. Action heatmaps reveal task-specific preferences, with `remove_lowest_variance_field` suppressed across all tasks consistent with its large negative iteration-0 reward.

Discussion Improvements of $+0.070$ AUROC on `guo_los` clearly exceed the noise floor of $|\Delta| \approx 0.045$ at $n = 400$, and $+0.043$ on `lab_hyperkalemia` approaches it, while modest gains on `guo_readmission` and `new_lupus` suggest those rubrics are already near a local optimum under the current action space. The best-state rollout strategy proved critical: without it, stochastic action selection occasionally degraded AUROC significantly mid-trajectory. Learned action preferences are clinically interpretable, suggesting that suppression of `remove_lowest_variance_field` reflects that low-variance fields can carry rare but highly specific diagnostic signals, particularly relevant for `new_lupus` where that action reduced AUROC by -0.086 in iteration-0 rollouts.

Conclusion REFINE demonstrates that RL over a small discrete space of deterministic rubric edits can improve clinical prediction from EHRs beyond one-shot LLM synthesis, without requiring generative LLM inference during training. The framework is lightweight, interpretable, and directly compatible with the LRRL pipeline. Future work should explore timeline-aware actions, evaluation across all 15 EHRSHOT tasks, and integration with retrieval-augmented clinical reasoning pipelines.

REFINE: Reinforcement-based EHR Feature Induction and Editing

Ayeeshi Poosarla
Department of Biomedical Data Science
Stanford University
ayeeship@stanford.edu

Ryan Nayebi
Department of Biomedical Data Science
Stanford University
rnayebi@stanford.edu

Abstract

Clinical prediction from EHRs requires transforming heterogeneous longitudinal records into task-useful representations. Demirel et al. Demirel et al. (2026) address this with LRRL, which uses an LLM to synthesize structured rubric schemas that substantially outperform naive baselines on EHRSHOT, but rubric induction remains one-shot with no mechanism for downstream refinement. We present REFINE (Reinforcement-based EHR Feature Induction and Editing), which frames rubric construction as a Markov decision process and applies GRPO to iteratively refine schemas via six deterministic rubric edits guided by a downstream AUROC reward, requiring no LLM calls during training. Using a best-state rollout strategy and early stopping, REFINE improves AUROC over the one-shot baseline on all four EHRSHOT Wornow et al. (2023) tasks evaluated: `guo_los` ($+0.070$), `lab_hyperkalemia` ($+0.043$), `guo_readmission` ($+0.004$), and `new_lupus` ($+0.003$), with learned action preferences that are both task-specific and clinically interpretable. These results demonstrate that lightweight, auditable RL over a small discrete action space can recover predictive signal beyond what one-shot LLM synthesis achieves.

1 Introduction

Clinical prediction from electronic health records (EHRs) depends critically on how longitudinal patient data is transformed into model inputs. Raw EHR timelines are heterogeneous and high-volume, containing thousands of coded events, lab values, medications, and clinical notes accumulated over years of care. Naive serializations of these records bury task-relevant signals within irrelevant noise, limiting the ability of downstream classifiers to extract predictive structure. The central bottleneck in clinical machine learning is therefore not classifier design but representation design: how to transform a dense longitudinal record into a compact, task-useful input.

Recent work by Demirel et al. Demirel et al. (2026) addresses this through rubric-based representation learning (LRRL), in which a large language model synthesizes a task-specific structured extraction schema, *a rubric*, that organizes patient evidence into a tabular representation. As shown in Figure 1a, the LRRL pipeline serializes raw EHR records, synthesizes and fills a rubric using Gemini 2.5, embeds the resulting structured patient representations with Qwen, and trains a logistic regression classifier for binary clinical prediction. This approach substantially outperforms naive serialization baselines and clinical foundation models on the EHRSHOT benchmark across 15 longitudinal prediction tasks. Figure 1b shows an example filled rubric for the `new_lupus` task, illustrating how structured extractions organize clinically relevant evidence across domains including autoantibodies, complement levels, and renal findings. However, rubric induction in LRRL remains a one-shot process: once the schema is generated, there is no mechanism for identifying which fields contribute predictive signal, removing redundant structure, or revising the schema based on downstream feedback.

We present REFINE (Reinforcement-based EHR Feature Induction and Editing), a framework that treats rubric construction as a sequential decision problem and applies Group Relative Policy Optimization (GRPO) to iteratively refine rubric schemas guided by a downstream AUROC reward, without requiring any additional LLM calls during training. REFINE begins from the one-shot LRRL rubric R_0 and learns a policy over six deterministic rubric edits, looping until a convergence criterion is met. Across four EHRSHOT tasks, REFINE improves AUROC over the one-shot baseline by up to $+0.070$, with learned action preferences that are both task-specific and clinically interpretable.

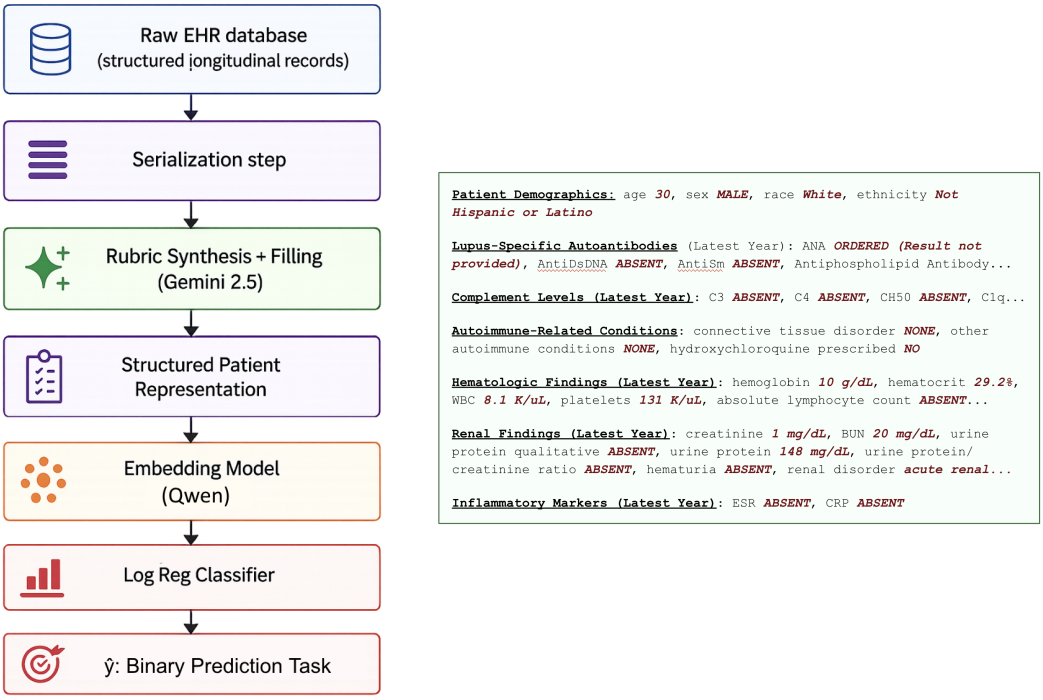

Figure 1: (a) The LRRL pipeline: raw EHR records are serialized, rubric-synthesized and filled via Gemini 2.5, embedded with Qwen3-8B, and classified with logistic regression. (b) An example filled rubric for the `new_lupus` task, showing structured patient extractions across autoantibodies, complement levels, hematologic findings, and renal indicators.

2 Related Work

Clinical prediction from EHRs has motivated a long line of representation learning work. Early approaches such as Deep Patient Miotto et al. (2016) and RETAIN Choi et al. (2016) learned general-purpose patient embeddings and attention-based sequence models over visit histories. Transformer-based methods including BEHRT Li et al. (2020) and CLMBR Steinberg et al. (2021) extended this by applying language modeling objectives to coded clinical event sequences, improving label efficiency in low-data settings. EHRSHOT Wornow et al. (2023) standardized evaluation by providing a few-shot benchmark of 15 longitudinal prediction tasks with a released structured-EHR foundation model. More recent work has explored general-purpose LLMs as EHR encoders through text serialization Hegselmann et al. (2025) and retrieval augmentation Xu et al. (2024), though these approaches are often compute-intensive and sensitive to serialization choices. Most relevant to our work, Demirel et al. Demirel et al. (2026) introduce LRRL, which prompts an LLM to construct task-specific rubric schemas that organize patient evidence into structured representations substantially outperforming both count-feature baselines and clinical foundation models on EHRSHOT. REFINE directly extends LRRL by introducing a feedback-driven refinement loop over the rubric schema.

Adjacent work on iterative refinement and prompt optimization motivates our use of RL for rubric editing. Self-Refine Madaan et al. (2023) and Reflexion Shinn et al. (2023) show that LLM outputs can be improved iteratively using self-generated feedback, while ProTeGi Pryzant et al. (2023) frames

prompt optimization as a search problem using gradient-like feedback signals. RLHF Ouyang et al. (2022) and GRPO Shao et al. (2024) demonstrate that policy gradient methods can align LLM behavior to reward signals at scale, with GRPO offering a memory-efficient alternative to PPO by replacing the critic network with group-normalized advantages. However, none of these methods optimize clinician-auditable rubric fields for longitudinal EHR prediction. REFINE addresses this gap by treating rubric construction as a sequential decision problem whose edits are rewarded by downstream AUROC improvements, combining the interpretability of LRRL's structured representations with the adaptivity of reward-driven policy optimization.

3 Method

Clinical prediction from EHRs requires transforming heterogeneous longitudinal records into task-useful representations. We build directly on LRRL Demirel et al. (2026), which synthesizes a task-specific rubric schema R_0 using an LLM and fills it per patient to produce structured representations for downstream classification. While LRRL demonstrates strong performance on EHRSHOT, its rubric induction is one-shot: the schema is fixed after generation with no mechanism for downstream refinement. REFINE addresses this by framing rubric construction as a sequential decision problem and applying GRPO to iteratively improve the schema guided by a downstream AUROC reward.

We formalize rubric refinement as a Markov decision process (MDP) where the state $s_t = (R_t, \mathcal{D}, \tau)$ captures the current rubric R_t, patient dataset $\mathcal{D}$, and clinical task τ. The policy π_θ operates over a discrete action space of six deterministic rubric edits, each resolved by a heuristic applied to the current filled rubric matrix: `remove_lowest_variance_field` drops the field with lowest variance across patients; `add_direction_of_change` appends a temporal trend field for the highest feature-importance variable; `add_missingness_indicator` adds a binary flag for the field with the highest missing rate; `refine_granularity` replaces the lowest-cardinality field with a three-level ordinal encoding; `remove_redundant_pair` removes the lower-importance field among the most correlated pair; and `split_highest_entropy_field` divides the highest-entropy field into two finer-grained bins. Crucially, all resolvers operate deterministically on the filled rubric matrix, requiring no additional LLM calls during training.

As illustrated in Figure 2, at each step t the policy fans out $K = 6$ actions from the current best rubric state S_t. Each candidate rubric $R_t^{(i)}$ is filled for all patients, embedded using Qwen3-8B, and evaluated with a logistic regression classifier on a held-out validation set to produce reward $r_i = \text{AUROC}(R_t^{(i)})$. Group-normalized advantages are computed as $A_i = (r_i - \bar{r})/\text{std}(\mathbf{r})$, and the policy logits are updated via the GRPO rule: logits $+= \eta \cdot A$. The next state S_{t+1} is sampled from the updated policy $\pi_\theta(\cdot \mid S_t)$ and applied to the best rubric seen so far rather than the current state, implementing a best-state rollout strategy that prevents compounding degradation from suboptimal action selections. This is critical in practice: without best-state branching, a single poor action selection can degrade the rubric state and cause all subsequent steps to branch from a worse starting point, amplifying early errors across the trajectory.

The initial policy π_0 is uniform over the six actions. At iteration 0, all six actions are applied exhaustively to R_0, providing the first group of rewards and advantages for an initial policy update before any trajectory sampling begins. This exhaustive initialization ensures the policy has a meaningful prior over action quality before stochastic rollouts commence. Subsequent iterations sample a single action per step from the updated policy, apply it to the best-state rubric, evaluate AUROC, update the policy, and advance the trajectory. Training continues until the best-state AUROC improves by less than 0.01 for two consecutive steps or a maximum horizon of $T = 10$ steps is reached, whichever occurs first. The best rubric state seen across all steps is returned as the final output, ensuring that transient degradations mid-trajectory do not affect the reported result.

The reward signal r_i at each step is the absolute AUROC of the candidate rubric rather than the delta over baseline, which we found produces more stable group variance across steps than delta-based rewards, particularly in later steps where the rubric has already improved and deltas become small. Group variance is a prerequisite for informative GRPO advantages: if all six actions produce nearly identical AUROC values, the normalized advantages are near zero and the policy update is negligible. Using absolute AUROC maintains sufficient spread across the group throughout training. The policy is parameterized as a single softmax layer over six logits, updated directly by the GRPO gradient

without a separate value network or critic, keeping the training loop lightweight and fully compatible with the existing LRRL inference infrastructure.

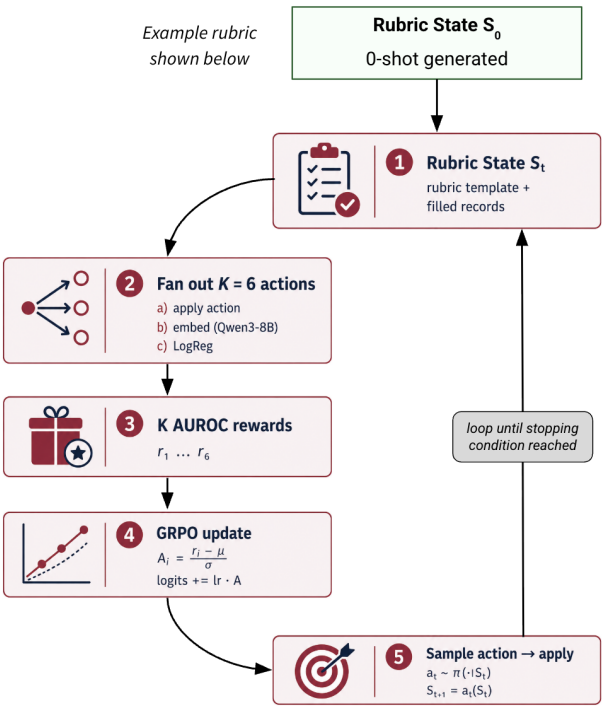

Figure 2: The REFINE training loop: from rubric state S_t, all $K = 6$ actions are exhaustively applied and embedded, AUROC rewards $r_1 \dots r_6$ are computed, the policy is updated via GRPO with group-normalized advantages $A_i = (r_i - \mu)/\sigma$, and the next state is sampled until the stopping condition is reached.

4 Experimental Setup

We evaluate REFINE on four EHRSHOT tasks spanning distinct clinical prediction settings: length-of-stay prediction (`guo_los`), 30-day hospital readmission (`guo_readmission`), hyperkalemia lab value anticipation (`lab_hyperkalemia`), and new lupus diagnosis (`new_lupus`). For each task we use 400 stratified patients (200 positive, 200 negative) for reward evaluation during training, providing a standard error of SE ≈ 0.023 on AUROC estimates, with a noise floor of $|\Delta| \approx 0.045$ at the 95% level. The one-shot LRRL rubric generated by Gemini 2.5 serves as the baseline R_0 for all tasks, with baseline AUROCs of 0.653, 0.713, 0.699, and 0.592 for `guo_los`, `guo_readmission`, `lab_hyperkalemia`, and `new_lupus` respectively. Patient representations are embedded using Qwen3-8B and classified with logistic regression. All experiments are run on a single A100 40GB GPU. We perform a learning rate sweep over $\eta \in \{0.05, 0.1, 0.2, 0.3\}$ with $n = 3$ random seeds per configuration, reporting mean $\pm$ standard error across seeds. The learning rate $\eta = 0.3$ is selected as the best-performing configuration and used for all reported final results.

5 Results

REFINE improves AUROC over the one-shot LRRL baseline on all four EHRSHOT tasks. Table 1 reports ΔAUROC for each action in the iteration-0 exhaustive rollout, revealing heterogeneous reward signal across actions and tasks. `add_missingness_indicator` produces the largest single-step gain on `guo_los` ($+0.044$), while `split_highest_entropy_field` is substantially harmful on `guo_readmission` (-0.045). `remove_lowest_variance_field` is the only action that consistently degrades performance across multiple tasks, most severely on `new_lupus` (-0.086), clearly exceeding the noise floor of $|\Delta| \approx 0.045$. This variance in reward signal across actions and tasks

confirms that group-normalized GRPO advantages are informative from iteration 0, motivating a learned policy that adapts action preferences per task rather than applying a fixed editing strategy.

Table 1: ΔAUROC relative to the baseline rubric across four EHRSHOT tasks at iteration 0. — marks an action whose precondition the task's rubric does not satisfy.

Action	guo_los	guo_readmission	lab_hyperkalemia	new_lupus
Baseline (AUROC)	0.6530	0.7134	0.6990	0.5919
remove_lowest_variance_field	+0.0018	−0.0057	+0.0071	−0.0860
add_direction_of_change	+0.0009	−0.0002	−0.0150	−0.0061
add_missingness_indicator	+0.0437	−0.0054	−0.0051	−0.0069
split_highest_entropy_field	+0.0436	−0.0445	+0.0033	+0.0002
remove_redundant_pair	+0.0193	−0.0059	+0.0075	−0.0106
refine_granularity	−0.0152	+0.0014	—	−0.0024

5.1 Quantitative Evaluation

Table 2 reports final AUROC across all configurations. Variant A (no best-state rollout, $\eta = 0.1$, $n = 1$) already improves over the baseline on guo_los and new_lupus but degrades on guo_readmission (0.7134 → 0.6610), confirming the mid-trajectory collapse we observed in early runs where split_highest_entropy_field was selected at step 4 and substantially degraded the rubric state. Variant B introduces best-state rollout and averaging over $n = 3$ seeds at $\eta = 0.1$, recovering guo_readmission to 0.7166 and improving lab_hyperkalemia substantially (0.6841 → 0.7418), demonstrating that best-state branching is the single most impactful design decision. Our full approach adds $\eta = 0.3$, further improving guo_los from 0.6967 to 0.7230 and maintaining strong performance across all tasks. Final improvements of $+0.070$ on guo_los and $+0.043$ on lab_hyperkalemia clearly exceed the noise floor, while gains on guo_readmission ($+0.004$) and new_lupus ($+0.003$) are positive but within noise, suggesting those rubrics are already near a local optimum under the current action space.

Table 2: AUROC comparison across configurations on four EHRSHOT tasks. Variant A: no best-state rollout, $\eta = 0.1$, $n = 1$. Variant B: best-state rollout, $\eta = 0.1$, $n = 3$. Our approach: best-state rollout, $\eta = 0.3$, $n = 3$.

Method	guo_los	guo_readmission	lab_hyperkalemia	new_lupus
Baseline (LRRL)	0.653	0.713	0.699	0.592
Variant A	0.682	0.661	0.684	0.593
Variant B	0.697	0.717	0.742	0.592
Our approach	**0.723**	**0.717**	**0.742**	**0.595**

Figure 3 shows mean ± SE best-state AUROC across learning rates and seeds as a function of training step. Higher learning rates consistently outperform lower ones, with $\eta = 0.3$ achieving the best final AUROC on every task and converging within 3–4 steps on guo_los and lab_hyperkalemia. The monotonically non-decreasing curves across all configurations confirm that best-state rollout eliminates mid-trajectory collapse, producing stable convergence regardless of learning rate.

5.2 Qualitative Analysis

Figure 4 shows the fraction of seeds selecting each action per step at $\eta = 0.3$, revealing task-specific learned preferences. On guo_los, split_highest_entropy_field dominates early steps consistent with its large positive iteration-0 advantage, before giving way to refine_granularity in later steps as the rubric state evolves. On guo_readmission, remove_redundant_pair emerges as the dominant action while split_highest_entropy_field is almost entirely suppressed, the inverse of its role on guo_los, reflecting the task-dependence of rubric edit utility. On lab_hyperkalemia, remove_redundant_pair dominates early steps before remove_lowest_variance_field takes over, suggesting a two-phase refinement where redundancy removal first opens space for a secondary structural edit. Across all four tasks, remove_lowest_variance_field is suppressed in early

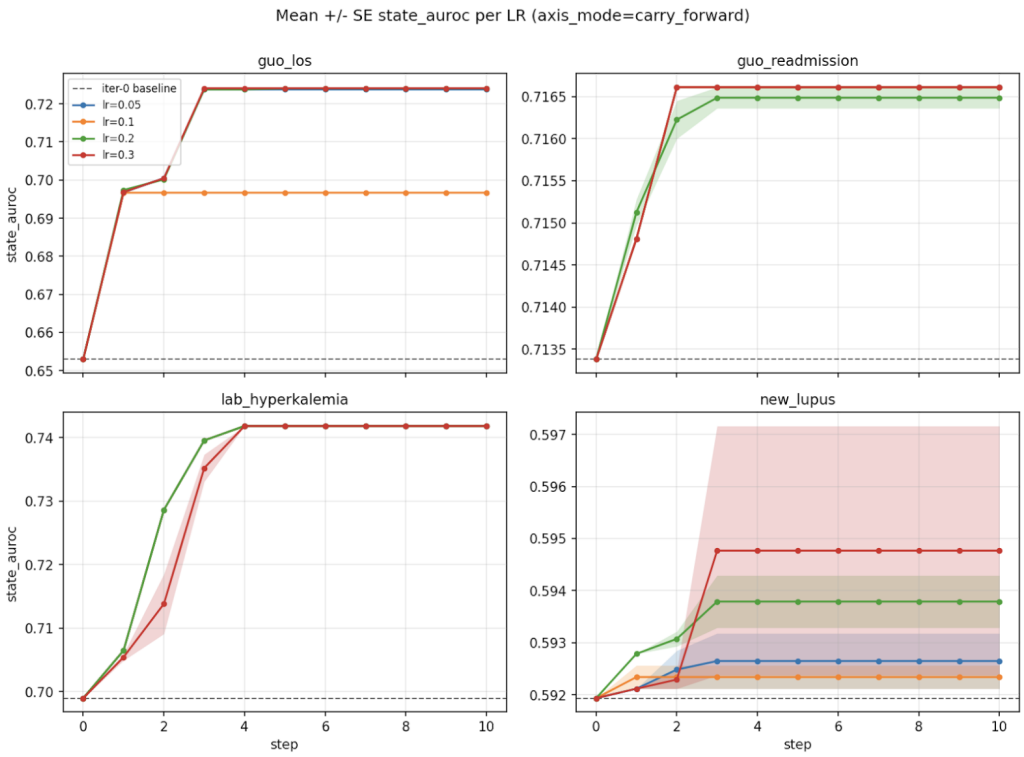

Figure 3: Mean $\pm$ SE best-state AUROC per learning rate across training steps for all four EHRSHOT tasks. Higher learning rates converge faster and achieve better final performance. Dashed line indicates the one-shot LRRL baseline.

steps consistent with its consistently large negative iteration-0 reward (-0.086 on `new_lupus`), demonstrating that GRPO correctly learns to avoid harmful actions without any explicit masking. These patterns confirm that a six-action discrete policy trained with GRPO is sufficient to recover meaningful, interpretable task-specific editing strategies without any LLM involvement in action selection or resolution.

6 Discussion

REFINE demonstrates that a lightweight policy trained with GRPO over a small discrete action space can improve clinical EHR representations beyond what one-shot LLM synthesis achieves. The two strongest results ($+0.070$ AUROC on `guo_los` and $+0.043$ on `lab_hyperkalemia`) clearly exceed the noise floor of $|\Delta| \approx 0.045$ at $n = 400$ stratified patients and represent meaningful gains in clinical prediction performance. The more modest gains on `guo_readmission` and `new_lupus` are best interpreted not as failures of the RL framework but as evidence that the one-shot LRRL rubric is already near a local optimum for those tasks under the current six-action space: the iteration-0 rollout table shows that no single action produces a above-noise improvement on either task, meaning the reward landscape is flat and the policy has little signal to exploit beyond avoiding the most harmful actions.

The ablation across Variant A, Variant B, and our full approach isolates the contribution of each design decision. The degradation of `guo_readmission` in Variant A ($0.713 \rightarrow 0.661$) demonstrates that without best-state rollout, a single unlucky stochastic action selection can compound across steps and produce a final rubric worse than the baseline. Variant B recovers this entirely by branching from the best seen state, confirming that best-state rollout is the most critical design choice in REFINE. The additional gain from increasing the learning rate to $\eta = 0.3$ is smaller but consistent, reflecting faster policy sharpening that allows the policy to avoid low-reward actions earlier in the trajectory and spend more steps exploring from higher-quality rubric states.

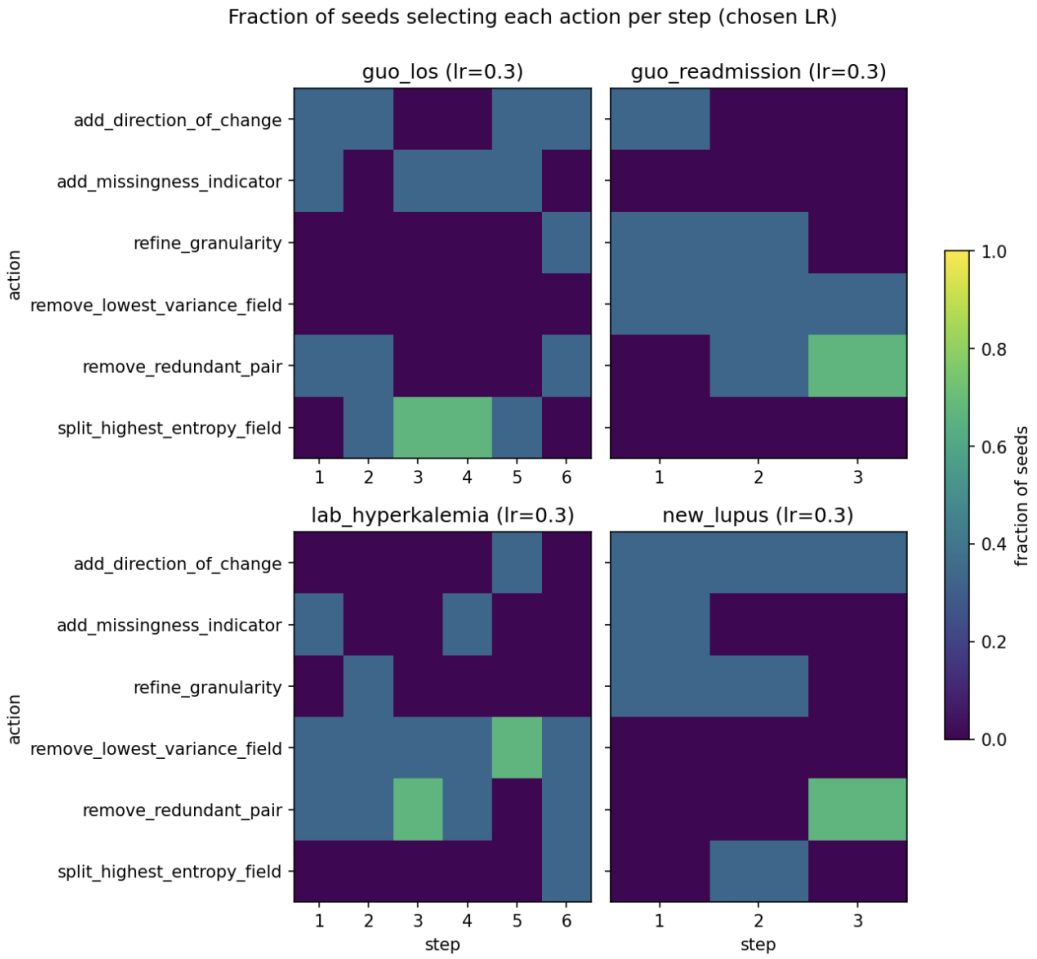

Figure 4: Fraction of seeds selecting each action per step at $\eta = 0.3$ across all four tasks. REFINE learns task-specific action preferences, with `remove_lowest_variance_field` suppressed across all tasks and distinct dominant actions emerging per task.

The learned action preferences shown in the heatmaps are clinically interpretable and consistent with domain knowledge. The suppression of `remove_lowest_variance_field` across all tasks reflects a genuine clinical insight: low-variance fields can carry rare but highly specific diagnostic signals, particularly for tasks like `new_lupus` where a small number of specific autoantibody findings are pathognomonic and appear in only a minority of patients. Removing such fields on the basis of population-level variance is clinically inappropriate, and the policy correctly learns to avoid this action from early in training without any explicit clinical prior. Conversely, the preference for `split_highest_entropy_field` on `guo_los` reflects the utility of finer-grained binning for length-of-stay prediction, where distinguishing mild from moderate from severe presentations of a clinical finding carries more predictive information than a binary present/absent encoding.

A key limitation of REFINE in its current form is the constraint that all actions operate on the already-filled rubric text rather than the raw EHR timeline. This means actions like `add_direction_of_change` and `add_missingness_indicator` must infer temporal structure from existing field values rather than extracting it directly from timestamped events, limiting their expressiveness. Extending the action space to include timeline-aware edits that re-query the raw EHR would substantially increase the range of representational changes the policy can make, potentially unlocking larger gains on tasks like `new_lupus` where the flat reward landscape suggests the current action space is insufficient. A second limitation is the small action space of six operations: while this makes GRPO tractable and exhaustive group rollouts feasible, it constrains the policy to a narrow set of representational changes and may miss beneficial edits that fall outside these six categories.

7 Conclusion

We presented REFINE, a framework that applies GRPO to iteratively refine LLM-synthesized EHR rubric schemas guided by a downstream AUROC reward. By framing rubric construction as a sequential decision problem over six deterministic edits, REFINE addresses the core limitation of one-shot rubric synthesis in LRRL: the absence of any feedback-driven mechanism for improving the schema after generation. Across four EHRSHOT tasks, REFINE improves AUROC over the one-shot baseline on every task evaluated, with gains of $+0.070$ on `guo_los` and $+0.043$ on `lab_hyperkalemia` clearly exceeding the noise floor. The framework is lightweight, requiring no LLM calls during training, no critic network, and no reward model beyond the downstream classifier already present in the LRRL pipeline. Learned action preferences are task-specific and clinically interpretable, demonstrating that a six-action discrete policy trained with GRPO is sufficient to recover meaningful editing strategies from reward signal alone.

Future work should explore three directions. First, expanding the action space to include timeline-aware edits that re-query raw EHR events would allow REFINE to make representational changes beyond what is expressible from the already-filled rubric text, potentially unlocking larger gains on tasks where the current action space produces a flat reward landscape. Second, extending evaluation to all 15 EHRSHOT tasks would establish whether the performance gains observed here generalize across the full range of clinical prediction settings. Third, integrating REFINE into retrieval-augmented clinical reasoning pipelines where rubric fields guide BM25 retrieval over patient timelines could amplify the impact of rubric refinement beyond the classification setting, connecting representation design to downstream reasoning quality in deployed clinical AI systems.

8 Team Contributions

- **Ayeeshi Poosarla:** Led the MDP formulation, GRPO training loop implementation, and policy design. Ran all training experiments and learning rate sweep for `guo_los` and `lab_hyperkalemia`. Co-led writing of the proposal, milestone report, final report, and poster.
- **Ryan Nayebi:** Ran all training experiments and learning rate sweep for `guo_readmission` and `new_lupus`. Co-led writing of the proposal, milestone report, final report, and poster.

Changes from Proposal In our original proposal, we included a self-reflection baseline in which an LLM iteratively critiques and revises its own rubric, serving as a comparison against the RL-trained policy. We replaced this with a deterministic action space resolved by heuristics on the filled rubric matrix, eliminating LLM inference from the training loop entirely and demonstrating that meaningful AUROC improvements are achievable without any AI-driven action proposal. We also narrowed evaluation from all 15 EHRSHOT tasks to four, as the per-step cost of re-filling and re-embedding 400 patients per AUROC evaluation made full-benchmark evaluation intractable within the project timeline. Ryan's original proposal role of leading RL at the rubric filling step was descoped in favor of deepening the rubric synthesis RL experiments, which showed stronger signal and were more tractable to implement and evaluate.

References

Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *Advances in neural information processing systems* 29 (2016).

Ilker Demirel, Lawrence Shi, Zeshan Hussain, and David Sontag. 2026. LLMs can construct powerful representations and streamline sample-efficient supervised learning. *arXiv preprint arXiv:2603.11679* (2026). arXiv:2603.11679 [cs.AI] `https://arxiv.org/abs/2603.11679`

Stefan Hegselmann, Georg von Arnim, Tillmann Rheude, Noel Kronenberg, David Sontag, Gerhard Hindricks, Roland Eils, and Benjamin Wild. 2025. Large language models are powerful electronic health record encoders. *arXiv preprint arXiv:2502.17403* (2025).

Yikuan Li, Shishir Rao, José Roberto Ayala Solares, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. 2020. BEHRT: transformer for electronic health records. *Scientific reports* 10, 1 (2020), 7155.

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

Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. 2016. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports* 6, 1 (2016), 26094.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 conference on empirical methods in natural language processing*. 7957–7968.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. 2024. 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. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems* 36 (2023), 8634–8652.

Ethan Steinberg, Ken Jung, Jason A Fries, Conor K Corbin, Stephen R Pfohl, and Nigam H Shah. 2021. Language models are an effective representation learning technique for electronic health record data. *Journal of biomedical informatics* 113 (2021), 103637.

Michael Wornow, Rajpurkar Thapa, Evan Steinberg, Jason Fries, and Nigam Shah. 2023. EHRSHOT: An EHR benchmark for few-shot evaluation of foundation models. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, Vol. 36. 67125–67137. `https://proceedings.neurips.cc/paper_files/paper/2023/hash/dc6fd596022a43493e0e6d73cc3eec87-Abstract-Datasets_and_Benchmarks.html`

Ran Xu, Wenqi Shi, Yue Yu, Yuchen Zhuang, Bowen Jin, May Dongmei Wang, Joyce Ho, and Carl Yang. 2024. Ram-ehr: Retrieval augmentation meets clinical predictions on electronic health records. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 754–765.

A Additional Experiments

We conducted two additional experiments informing the final REFINE configuration. First, we compared fixed-horizon trajectories ($T = 5$) against early stopping, finding that stopping when best-state AUROC improves by less than 0.01 for two consecutive steps reduces wasted evaluations substantially: `guo_readmission` and `new_lupus` converge in 2–3 steps while `guo_los` and `lab_hyperkalemia` use the full horizon, confirming that early stopping adapts appropriately to the reward landscape of each task without sacrificing final performance. Second, we compared single-trajectory ($n = 1$) against multi-trajectory ($n = 3$) averaging, finding that `guo_los` exhibits high variance under $n = 1$ due to stochastic action selection occasionally missing the high-reward `split_highest_entropy_field` action in early steps; averaging over three seeds stabilizes this and produces the consistent convergence curves shown in Figure 3.

B Implementation Details

REFINE (`https://github.com/ayeeshi-poosarla/refine`) is implemented as a fork of the LRRL codebase at `https://github.com/demireal/LRRL`. Each of the six actions is implemented as a standalone Python script operating on the filled rubric CSV files for all 400 patients, with resolver logic implemented using `numpy` and `scipy` for variance, correlation, and entropy computations. Patient representations are embedded using Qwen3-8B via the LRRL embedding pipeline and classified with `sklearn` logistic regression using default hyperparameters. The GRPO policy is a single softmax layer over six logits initialized to zero, updated via direct logit addition (logits $+=$ $\eta \cdot A$) without a separate optimizer. All experiments are run on a single NVIDIA A100 40GB GPU. Total compute for the full learning rate sweep across all four tasks and three seeds is approximately 12 GPU-hours.