REMIX: ONLINE EVALUATION OF EVOLVING MEDICAL AI SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Medical AI systems built on large language models can change faster than evidence about their performance accumulates. How can we evaluate a new version when only a few patient outcomes are available? We introduce ReMix, an adaptive mixture of historical-borrowing and direct benchmark-score estimators. At each update, paired benchmark evaluations calibrate borrowing in replay-informed estimators, while the new version's benchmark scores provide direct estimates. ReMix adjusts their weights as patient outcomes arrive. We bound its cumulative squared prediction loss relative to the best constituent estimator for any reward sequence. In offline experiments using real patient questions from MedHELM, ReMix reduces mean squared estimation error by 39% after just five post-update patients and cumulative squared estimation error by 29%, compared with evaluating each version from scratch. It also achieves lower average cumulative error than methods for borrowing historical information in clinical trials on both testbeds.

1 INTRODUCTION

Large language models (LLMs) are moving from benchmarks into clinical practice, drafting replies to patient messages (Garcia et al., 2024), generating clinical notes from doctor–patient conversations (Tierney et al., 2024), and assisting during primary-care visits (Korom et al., 2025). However, LLM-based medical AI systems often rely on third-party foundation models that are regularly updated and deprecated. A review of 4,609 clinical LLM studies found that 87.7% of the models evaluated were closed-source (Chen et al., 2026); and a survey of 36 U.S. clinical and translational research institutions found that 86.1% used proprietary third-party LLMs (Idnay et al., 2025). Clinical developers do not control when these models are updated or retired. Between January and September 2026, Anthropic released at least ten new Claude models and retired seven older ones (Anthropic, 2026a;b), while OpenAI retired at least seven GPT model versions from its API (OpenAI, 2026). Each update can alter performance on the population the AI system serves, and the direction of change need not agree across evaluation settings. Per-instance results released with MedHELM (Bedi et al., 2026) show how far these changes can diverge. Moving from Gemini 2.0 Flash to Gemini 2.5 Pro raised accuracy on MedQA (Jin et al., 2021) by 11 percentage points but lowered the score on real patient questions from iCliniq (Zeng et al., 2020) by 14 points. Performance on the deployment population must therefore be continuously monitored as the system evolves.

Such non-stationarity poses new statistical challenges for the postmarket evaluation of clinical AI. Conventional clinical studies accumulate evidence about a prespecified intervention, and reporting guidelines for trials of AI interventions ask investigators to state which version of the algorithm was used (Liu et al., 2020). Standard LLM benchmarks likewise evaluate specified model versions on fixed test sets (Chang et al., 2024; Bedi et al., 2025; Zhao et al., 2026). In contrast, a deployed medical agent may itself change while evidence about its performance is still accumulating. The U.S. Food and Drug Administration (FDA) has recently highlighted the need for timely evaluation of such updates. A recent FDA discussion paper notes that such AI devices may undergo "unplanned changes arising from updates to an underlying third-party foundation model," and asks "How can a manufacturer detect, evaluate, and respond to such changes in a timely manner?" and "How might the extent of re-benchmarking or other evidence be scaled to the nature and expected impact of a given modification?" (FDA, 2026, pp. 21, 30–31). The resulting problem is inherently sequential. As patient outcomes arrive and agent versions change, how can we accurately track the performance

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

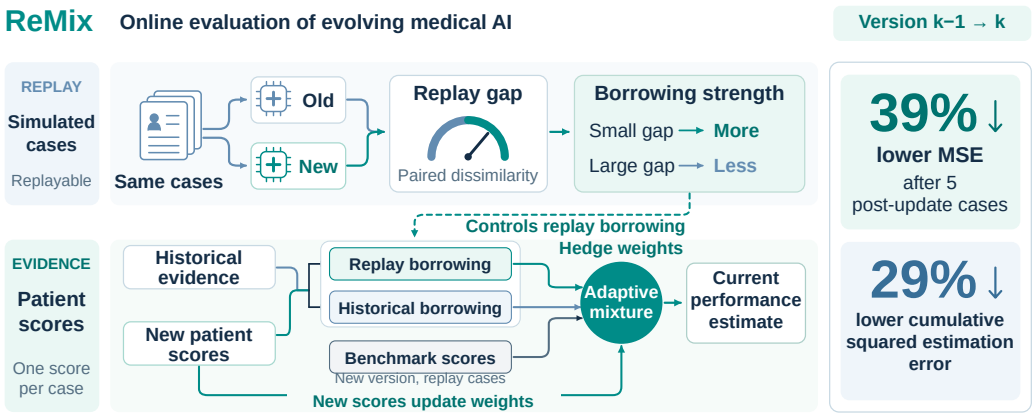

Figure 1: **ReMix tracks performance across model updates.** Paired replay on the same benchmark cases controls historical borrowing, while the new version's benchmark scores also provide direct estimates. ReMix combines these estimators with established historical-borrowing methods and updates their weights as patient outcomes arrive. Insets show mean-error reductions on the full MedHELM sequence relative to current-only evaluation.

of the version currently serving patients? To our knowledge, we are among the first to tackle the urgent and underexplored problem of sequential performance estimation for evolving LLM-based medical agents.

Evaluating each new version from scratch is noisy when few patient outcomes are available, while borrowing from earlier versions risks bias. Dynamic-borrowing methods assess compatibility using newly observed outcomes (Hobbs et al., 2011; Chu & Yi, 2021), leaving limited guidance immediately after an update. We use paired replay on simulated benchmarks to measure how much the agent changed before new patient outcomes arrive: the old and new versions are evaluated on the same simulated cases. ReMix combines established historical-borrowing methods, replay-informed borrowing estimators, and direct benchmark-score estimators. It adaptively adjusts their weights as patient outcomes arrive, using squared-loss online aggregation (Freund & Schapire, 1997; Vovk, 2001) (Figure 1).

Our main contributions are threefold:

1. We formalize online performance evaluation of medical AI under successive updates, where each patient is observed once and external benchmarks can be replayed across versions.
2. We propose ReMix, which combines replay-informed estimators with existing historical-borrowing methods and adapts their weights as patient outcomes arrive. We identify conditions under which replay improves estimation and use online-learning guarantees to relate the mixture's performance to its best constituent estimator.
3. Across two testbeds covering patient questions and interactive clinical tasks, ReMix improves estimation accuracy and reduces Total CVaR on the full model sequences. We evaluate its components through ablations and use effective current sample size to express sample-efficiency gains in units of additional post-update patients.

2 Related Work

Evaluation of medical LLMs and agents. Medical benchmarks have moved from exam-style questions to open-ended conversations, simulated patients, and tool use in electronic health records (Arora et al., 2025; Schmidgall et al., 2024; Jiang et al., 2025; Bedi et al., 2026), and recent work evaluates agents along their evolution (Zheng et al., 2026; Jiang et al., 2026). For deployed systems, dynamic deployments call for evaluation across updates (Rosenthal et al., 2025), and model-equality testing and drift attribution detect when a served model or its judge has changed (Gao et al., 2025;

Chen et al., 2024; Li, 2026). However, these works either evaluate a fixed model or detect a change without estimating its effect on patients. In contrast, our work estimates the current performance of each new version from its first patients onward.

Dynamic borrowing in clinical trials. Power priors, commensurate priors, and robust meta-analytic-predictive priors discount historical data according to their compatibility with current data (Ibrahim & Chen, 2000; Hobbs et al., 2011; Schmidli et al., 2014), and the discrepancy-weighted average of Chu & Yi (2021) uses the observed difference between historical and current means. Bayesian device trials can also borrow from computational models of virtual patients (Haddad et al., 2017). These methods measure compatibility on real data or treat simulation as additional observations. In contrast, our work measures compatibility before real data arrive, by replaying both versions on simulated cases.

Evaluation with proxies. Prediction-powered inference and AutoEval correct proxy predictions with labeled target data (Angelopoulos et al., 2023; Boyeau et al., 2025), tinyBenchmarks and Anchor Points select informative items (Polo et al., 2024; Vivek et al., 2024), and sim-to-real studies relate simulated to real performance (Kadian et al., 2020; Badithela et al., 2025; Luo et al., 2025). FAQ learns model and question factors from historical responses on a fixed question bank and actively selects questions for a new model (Wu et al., 2026). In our setting, each real patient is observed only once under the current version: historical responses are unavailable for future patients, and the evaluator cannot choose which patient arrives next. Thus, FAQ's question-factor transfer and active-query design are not directly available under our observation protocol. Our method uses replayable simulated cases to measure the size of an update and calibrate borrowing from historical patient outcomes.

Comparing model versions. Negative-flip and compatibility studies measure the new errors introduced by an update (Bansal et al., 2019; Srivastava et al., 2020; Yan et al., 2021; Echterhoff et al., 2024), and DISCERN certifies a no-regression verdict from disagreements on shared inputs (Balachandran, 2026). These methods compare two models on the same inputs. Our work uses such comparisons on simulated inputs to estimate performance on real patients, who are observed only once.

Online learning with expert advice. Hedge and the Aggregating Algorithm (Freund & Schapire, 1997; Vovk, 2001; Cesa-Bianchi & Lugosi, 2006), together with forecast combination (Stock & Watson, 2004; Smith & Wallis, 2009), provide the aggregation we use. We construct replay-based borrowing estimators and aggregate them to continuously estimate the performance of evolving medical agents.

3 Problem Formulation

Versions and updates. We consider a medical AI system deployed through successive versions $\pi_1, \ldots, \pi_K$. Version k begins serving patients at time t_k and remains active until $t_{k+1} - 1$, where $t_1 = 1$ and $t_{K+1} = T + 1$ for a stream of T patients. We write $k(t)$ for the version serving patient t. Updates are announced when they occur; the evaluator need not know future update times.

Patient observations. Each patient is evaluated once under the deployed version, yielding a reward $r_t \in [0, 1]$. The evaluator observes only this reward and cannot replay the patient under other versions. We hold the target population $\mathcal{P}$ and grading rule fixed, so that changes in performance reflect changes in the deployed system.

Replayable benchmarks. The evaluator also has access to M simulated or exam-style benchmarks. From benchmark m, we select J_m fixed replay cases indexed by $j = 1, \ldots, J_m$; J_m is the replay budget, not the full benchmark size. At each update, the new version can be evaluated on these cases before any new patient outcome is observed, and its rewards can be paired with stored rewards from the preceding version on the same cases. Benchmark populations and grading rules may differ from those of real patients.

Estimation objective. Our goal is to estimate the performance of the currently deployed version,

$$v^{(k)} = \mathbb{E}[r \mid \pi_k, \mathcal{P}], \qquad v_t = v^{(k(t))}. \tag{1}$$

After observing r_t, the evaluator reports $\hat{v}_t$ using patient outcomes and benchmark evaluations available by time t. We assess estimation accuracy through cumulative squared error,

$$\mathcal{E}_T = \sum_{t=1}^{T} (\hat{v}_t - v_t)^2. \tag{2}$$

The central challenge is deciding how much information from previous versions remains useful immediately after an update. Appendix A summarizes the main notation.

4 Replay-Informed Mixture of Estimators

ReMix combines established historical-borrowing methods with replay-informed borrowing estimators and direct benchmark-score estimators. It adaptively reweights these estimates as patient outcomes arrive, using their prediction losses to adjust the contribution of each estimator.

4.1 Replay-informed estimators

Within version k, let $h = t - t_k + 1$ and $L_k = t_{k+1} - t_k$. For $t = t_k + h - 1$, the global and within-version notation are related by

$$r_h^{(k)} = r_t, \qquad \hat{v}_{k,h} = \hat{v}_t, \qquad \tilde{v}_{k,h}(i) = \tilde{v}_t(i).$$

The same time mapping applies to the estimator and weight vectors. Write $\bar{r}_h^{(k)} = h^{-1} \sum_{q=1}^{h} r_q^{(k)}$ for the current mean, and $\mathcal{F}_{k,h}$ for the information available after h current outcomes, including historical outcomes and available benchmark evaluations. Thus $\mathcal{F}_{k,0}$ includes replay for the update but no current outcomes. We include one estimator of each type per benchmark. All estimators use the current sample mean on the initial version, before historical outcomes are available.

Dissimilarity-based borrowing. At update $k \geq 2$, benchmark m supplies paired rewards on the same cases, giving

$$d_k^{(m)} = \frac{1}{J_m} \sum_{j=1}^{J_m} \left| \tilde{r}_j^{(m,k)} - \tilde{r}_j^{(m,k-1)} \right|. \tag{3}$$

For estimator i using this benchmark, let $\mu_k(i)$ and $P_k(i)$ be its own estimate and working variance carried from the previous version. We combine history with current outcomes using

$$\begin{aligned} \tilde{v}_t(i) &= \lambda_{k,h}(i)\mu_k(i) + (1 - \lambda_{k,h}(i))\bar{r}_h^{(k)}, \\ \lambda_{k,h}(i) &= \frac{\hat{\sigma}_{k,h}^2/h}{\hat{\sigma}_{k,h}^2/h + P_k(i) + (d_k^{(m)})^2}. \end{aligned} \tag{4}$$

Here $\hat{\sigma}_{k,h}^2$ estimates current reward variance. The formula is a Gaussian working posterior mean: larger replay changes or greater historical uncertainty reduce borrowing. Appendix D.1 gives the variance estimates and state updates.

Direct benchmark-score estimate. The second estimator uses the new version's mean benchmark reward directly:

$$\tilde{v}_t(i) = \frac{1}{J_m} \sum_{j=1}^{J_m} \tilde{r}_j^{(m,k)}, \qquad k \geq 2. \tag{5}$$

This estimate stays fixed within the version and is useful when benchmark scores approximate target performance.

4.2 Adaptive aggregation of estimators

ReMix combines N online estimators using Hedge weights (Freund & Schapire, 1997), with estimator outputs $\tilde{\mathbf{v}}_t \in [0,1]^N$ and weights $\mathbf{w}_t$ (Algorithm 1).

Algorithm 1 ReMix: Replay-Informed Mixture of Estimators

Input: N online estimators; learning rate $\eta \in (0, 2]$.

Initialize $\mathbf{w}_0 \leftarrow \mathbf{1}/N$.
for patients $t = 1, \dots, T$ **do**
  Observe version $k(t)$.
  if $t = t_{k(t)}$ **then**
    Evaluate the new version on the selected replay cases and cache its rewards; reuse the preceding version's cached rewards for paired comparisons when $k(t) > 1$.
  end if
  Observe reward r_t; set $\boldsymbol{\ell}_t \leftarrow \mathbf{0}$.
  If $t > t_{k(t)}$, set $\boldsymbol{\ell}_t \leftarrow (r_t\mathbf{1} - \tilde{\mathbf{v}}_{t-1})^2$.
  $\mathbf{w}_t \leftarrow \mathrm{softmax}\Big(-\eta \sum_{s=1}^{t} \boldsymbol{\ell}_s\Big)$.
  $\tilde{\mathbf{v}}_t \leftarrow \mathrm{UpdateEstimators}\Big(r_t, k(t), \{\tilde{r}_j^{(m,k(t))}\}_{m,j}\Big)$.
  Report $\hat{v}_t$ by (6).
end for

Here $\mathbf{1}$ is the all-ones vector, and vector squares and exponentials act componentwise.

Estimator collection. We use $N = 6 + 2M$ estimators: the two replay-informed estimates for each of M sources, plus the weighted-average estimator of Chu & Yi (2021), robust meta-analytic-predictive priors (rMAP) (Schmidli et al., 2014), self-adapting mixture priors (SAM) (Yang et al., 2023), elastic MAP priors (EMAP) (Zhang et al., 2022), commensurate priors (CP) (Hobbs et al., 2012), and normalized power priors (NPP) (Shen et al., 2022). The same aggregation accepts any online estimator with outputs in $[0, 1]$.

Mixture estimate. We combine the updated estimates using the squared-loss substitution of Vovk's Aggregating Algorithm (Vovk, 2001):

$$\begin{aligned} g_t(\omega) &= -\frac{1}{\eta} \log\big\langle \mathbf{w}_t, \exp\big[-\eta(\omega\mathbf{1} - \tilde{\mathbf{v}}_t)^2\big]\big\rangle\,, \\ \hat{v}_t &= \mathrm{clip}_{[0,1]}\left[\frac{1 + g_t(0) - g_t(1)}{2}\right]. \end{aligned} \tag{6}$$

Here $\mathrm{clip}_{[0,1]}$ clips to $[0, 1]$.

Regret guarantee. Let $\hat{\ell}_t = (r_t - \hat{v}_{t-1})^2$ for $t > t_{k(t)}$ and $\hat{\ell}_t = 0$ at version onsets, matching the estimator losses in Algorithm 1. Write $\mathcal{R}_T$ for cumulative regret against the best estimator in hindsight.

Theorem 1 (Squared-loss aggregation; Vovk, 2001; Cesa-Bianchi & Lugosi, 2006)**.** *For any reward sequence with $r_t \in [0, 1]$ for all t, suppose that, for every $t > t_{k(t)}$ and estimator $i \in [N]$, the prediction $\tilde{v}_{t-1}(i) \in [0, 1]$ is formed before observing r_t. Then ReMix with learning rate $0 < \eta \le 2$ satisfies, for all $T \ge 1$,*

$$\mathcal{R}_T := \sum_{t=1}^{T} \hat{\ell}_t - \min_{i \in [N]} \sum_{t=1}^{T} \ell_t(i) \le \frac{\log N}{\eta}. \tag{7}$$

The comparator is one estimator rule across the stream, whose predictions may adapt to each update. Under a conditional-mean assumption, the regret bound transfers to estimation MSE with an additional K boundary term (Appendix E.3).

How replay improves the mixture. Accurate replay supports a borrowing estimator with lower MSE; the regret bound transfers its cumulative gains to ReMix.

Proposition 1 (Benefit of replay-informed borrowing)**.** *Fix a borrowing estimator i and update $k \ge 2$. Condition on history and replay, and write $\mu = \mu_k(i)$, $P = P_k(i)$, and $d = d_k^{(m)}$. Suppose current rewards are independent with mean $v^{(k)}$ and variance $\sigma^2 > 0$, and*

$$|\mu - v^{(k-1)}| \le \sqrt{P}, \qquad |v^{(k)} - v^{(k-1)}| \le d.$$

For $q = P + d^2 > 0$ and $n_0 = \sigma^2/q$, Equation (4) with $\hat{\sigma}^2_{k,h} = \sigma^2$ satisfies

$$\mathbb{E}[(\tilde{v}_{k,h}(i) - v^{(k)})^2 \mid \mathcal{F}_{k,0}] \le \frac{\sigma^2}{h} - \frac{\sigma^2 n_0^2}{h(h+n_0)^2} < \frac{\sigma^2}{h}, \qquad h \ge 1. \tag{8}$$

The gain depends on calibrated changes and historical uncertainty. Appendix E.4 gives the proof and an extension allowing replay error.

Corollary 1 (From borrowing gains to mixture gains)**.** *Under Theorem 1 and the conditional-mean assumption of Appendix E.3, suppose a fixed estimator i in the collection improves expected cumulative estimation error over current-only by $G_T(i) > 0$. Then*

$$\mathbb{E}\mathcal{E}_T \le \mathbb{E}\mathcal{E}_T^{\mathrm{curr}} - G_T(i) + \frac{\log N}{\eta} + K, \tag{9}$$

where $\mathcal{E}_T^{\mathrm{curr}}$ is current-only cumulative squared estimation error. Thus ReMix improves on current-only whenever $G_T(i) > \log N/\eta + K$.

Appendix E.4 proves the corollary and sums the gains from Proposition 1 across updates.

5 Experiments

We evaluate how accurately ReMix tracks performance after model updates, how historical borrowing changes sample efficiency, and which components account for the gains.

5.1 Datasets and benchmarks

MedHELM. Our main testbed evaluates performance tracking on real patient questions from MedDialog-HealthCareMagic, MedDialog-iCliniq, and MedicationQA, with per-instance scores from MedHELM (Bedi et al., 2026). The model sequence spans five vendors. Four external benchmarks provide replay signals: ACI-Bench, HeadQA, MedBullets, and MedCalc-Bench.

Supplementary Testbed. We assemble this supplementary testbed from additional model rollouts to test whether the gains extend beyond patient questions to health conversations and interactive clinical tasks. HealthBench (health conversations) (Arora et al., 2025), MedAgentBench (EHR tool-use tasks) (Jiang et al., 2025), and MedSP (simulated-patient interactions) (Liang et al., 2026) each serve as a target, using the other two benchmarks as external sources. Appendix B documents both testbeds' model sequences, score pools, and grading.

5.2 Evaluation protocol

Model updates. Table 1 uses the full chronological model sequence in each suite: 13 MedHELM versions with 50 patients per version, and six Supplementary Testbed versions with 30 patients per version. Model order and update positions are fixed. Appendix G varies the number of patients per version from 25 to 50 while retaining the full model sequences and all targets.

Patient and benchmark sampling. For Table 1, each target uses 500 independent patient replays, shared across methods. Patients are sampled without replacement from the pool scored for every version in the sequence; their stored rewards are revealed only under the deployed version. Source anchors and their stored scores are fixed across all replays, so the randomness is patient sampling. The full-pool mean defines v_t and is used only for evaluation. Appendix C specifies the conditioning and aggregation.

ReMix configuration. We use $J_m = 20$ fixed cases per replay source, giving $N = 14$ estimators for MedHELM and $N = 10$ for the Supplementary Testbed, with learning rate $\eta = 2$.

5.3 Metrics

We measure mean-squared estimation error, $\mathrm{MSE}(t) = \mathbb{E}[(\hat{v}_t - v_t)^2]$, where the expectation is taken over the randomness of the patient sequence, the model-update process, and simulated benchmark evaluations up to time t. In our experiments, we condition on the model-version sequence, update

times, selected replay cases, and their stored scores, and estimate the resulting conditional metrics from 500 patient replays. To assess large errors, we use conditional value-at-risk (CVaR). For a loss X and confidence level $\alpha \in (0, 1)$,

$$\mathrm{CVaR}_\alpha(X) = \min_{z \in \mathbb{R}} \left\{ z + \frac{1}{1-\alpha} \mathbb{E}[(X - z)_+] \right\}, \qquad (x)_+ = \max\{x, 0\},$$

which averages the worst $1 - \alpha$ fraction of losses (Rockafellar & Uryasev, 2002). We use $\alpha = 0.95$, applying CVaR to squared estimation error at each position and cumulative squared error up to time T. Lower MSE indicates more accurate performance estimates on average; lower CVaR indicates smaller errors in the upper tail of the error distribution.

Following the MSE-matching principle of Wiesenfarth & Calderazzo (2020), we also report effective current sample size, $\mathrm{ECSS}(h) = n^\star(h) - h$. Here h is the number of post-update patients observed, and $n^\star(h)$ is the current-only sample size matching the method's MSE, obtained by interpolating the empirical current-only error curve. ECSS expresses the gain in units of additional post-update patients. Appendix C gives the aggregation and matching details.

Reporting. Position @ h is the hth patient of each updated version, at global time $t_k + h - 1$. For a fixed target and model sequence, excluding the initial version, the update-averaged MSE is

$$\mathrm{MSE@}h = \frac{1}{K-1} \sum_{k=2}^{K} \mathrm{MSE}(t_k + h - 1). \tag{10}$$

Thus MSE@5 averages the MSE at $t_k + 4$ across updates. CVaR is likewise computed over patient replays within each update before averaging across updates. Total MSE and Total CVaR are $\mathbb{E}\mathcal{E}_T$ and $\mathrm{CVaR}_\alpha(\mathcal{E}_T)$, respectively, including the initial version. For both time ranges, we normalize each target's risk by the corresponding current-only risk, then average the three target ratios equally within each suite.

5.4 Baselines

We compare with **current-only**, which averages the current version's outcomes; **pooling**, which averages all observed outcomes; and a **sliding window** of 40 patients. We also evaluate the six historical-borrowing methods in Section 4.2 as standalone baselines: the weighted-average estimator of Chu & Yi (2021), rMAP, SAM, EMAP, CP, and NPP. Their predictions and settings are identical inside and outside ReMix. Appendix D.2 gives implementation details, and Appendix F isolates the contributions of the two replay estimators and the prior-work methods.

5.5 Results

ReMix estimates performance more accurately with fewer patients. ReMix has the lowest MSE among the reported methods on both full chronological sequences, both after five patients and up to time T (Table 1). After five patients, its MSE ratios of 0.610 on MedHELM and 0.652 on the Supplementary Testbed correspond to 39.0% and 34.8% lower mean squared error than current-only. Its ECSS gains of 3.14 and 2.58 mean that five observations match the accuracy of approximately 8.14 and 7.58 current-only observations. Figure 2 shows how this early advantage changes as patient outcomes accumulate. Total MSE ratios of 0.706 and 0.662 show that the average accuracy gains persist when error is accumulated across the complete sequence.

ReMix reduces the severity of large cumulative estimation errors. ReMix also has the lowest Total CVaR in both suites (Figure 3): ratios of 0.575 and 0.697 correspond to 42.5% and 30.3% reductions relative to current-only, averaged across targets. Thus the improvement extends to the average of the largest 5% of cumulative errors, providing evidence of robustness beyond lower mean error.

The gains depend on the evaluation horizon and model sequence. ReMix does not minimize tail error immediately after an update. After five patients, NPP has lower CVaR on MedHELM (0.309 versus 0.439), and the sliding window has lower CVaR on the Supplementary Testbed (0.464 versus 0.574). These methods better limit the severity of large errors at this early position, despite having higher Total CVaR. On the Supplementary Testbed open-model sequence, NPP also has lower Total

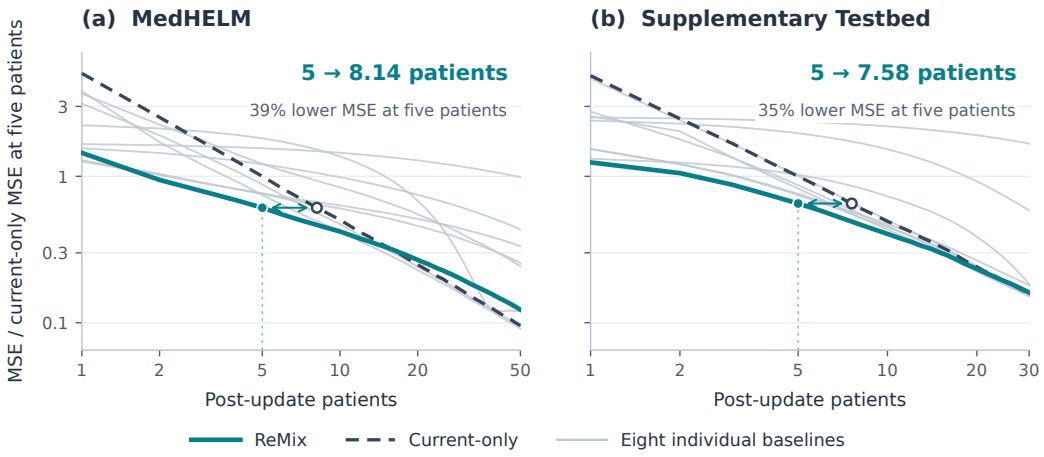

Figure 2: **ReMix improves accuracy early after an update.** MSE is averaged over patient replays and updated versions, normalized by each target's current-only MSE at five patients, then averaged across targets. Arrows match ReMix at five patients to the current-only sample size using the fitted ECSS reference curve. Gray lines show the eight baselines in Table 1. Both axes use logarithmic scales.

Table 1: **Main results on the full chronological model sequences.** MSE and CVaR are ratios to current-only; ECSS counts additional equivalent patients. Each suite averages its three target ratios. Total MSE and Total CVaR include the initial version. Bold marks the best displayed value.

	After 5 patients			Total	
Method	ECSS ↑	MSE ↓	CVaR ↓	MSE ↓	CVaR ↓
MedHELM: all-model sequence					
Pooling	-1.71	1.554	0.322	2.656	1.360
Sliding window (40)	-2.17	1.818	0.516	1.502	0.898
Chu & Yi (2021)	+1.67	0.747	0.613	0.816	0.784
rMAP	+1.50	0.766	0.376	1.144	0.730
SAM	+1.58	0.756	0.395	1.037	0.703
EMAP	-0.84	1.216	0.637	1.414	1.151
CP	+0.68	0.880	0.887	0.848	0.811
NPP	-0.82	1.210	**0.309**	1.619	0.918
ReMix (ours)	**+3.14**	**0.610**	0.439	**0.706**	**0.575**
Supplementary Testbed: all-model sequence					
Pooling	-2.87	2.369	0.637	2.805	1.984
Sliding window (40)	-0.10	1.024	**0.464**	0.926	0.860
Chu & Yi (2021)	+1.06	0.825	0.797	0.811	0.827
rMAP	+1.59	0.760	0.643	0.747	0.774
SAM	+1.72	0.744	0.669	0.721	0.748
EMAP	-0.02	1.006	0.971	0.994	0.991
CP	+0.78	0.870	0.848	0.827	0.843
NPP	-2.37	1.968	0.585	1.801	1.369
ReMix (ours)	**+2.58**	**0.652**	0.574	**0.662**	**0.697**

MSE and Total CVaR than ReMix (0.530 versus 0.539 and 0.596 versus 0.605). The results therefore support improved average accuracy and cumulative tail risk on the full sequences, with tradeoffs at shorter horizons and on model subsets (Appendix F).

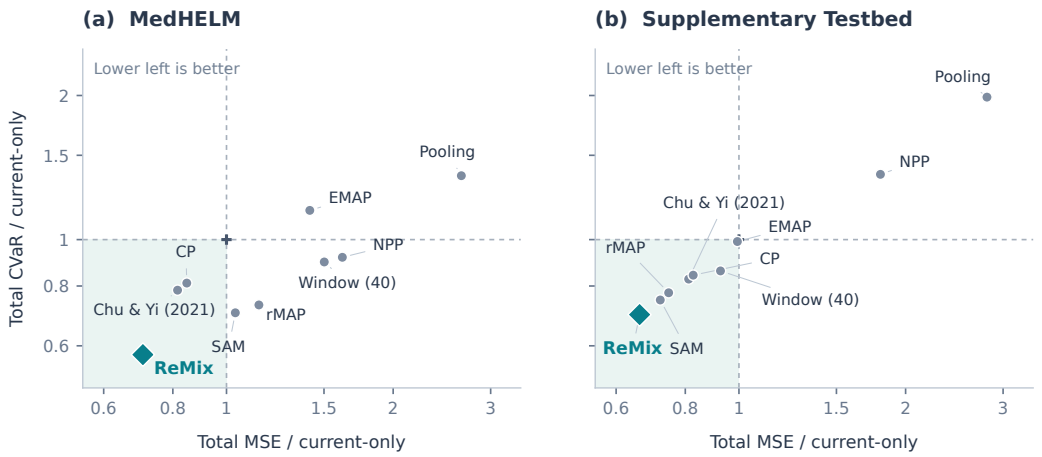

Figure 3: **Total MSE and Total CVaR.** Each point shows a method's Total MSE and Total CVaR, normalized by current-only within each target and averaged across targets. CVaR uses $\alpha = 0.95$. The shaded region improves both metrics over current-only; lower left is better. Both axes use logarithmic scales.

The accuracy gains persist across version lengths. Across 25–50 patients per version, ReMix retains the lowest mean whole-stream MSE among the compared methods in both suites. Its relative improvement over current-only is larger at shorter version lengths (Appendix G).

6 Conclusion

Discussion. ReMix uses benchmark replay to guide historical borrowing and provide direct score estimates, then combines these with existing borrowing methods through online aggregation. Its regret guarantee holds for any reward sequence. On the full MedHELM sequence, five patients after an update match the accuracy of 8.14 current-only patients, and cumulative mean error falls by 29%. Total CVaR also falls on both full sequences.

Our results suggest that evidence can accumulate across model updates, with the influence of earlier observations adapting as new outcomes arrive. This supports continuous evaluation of medical AI as its underlying models evolve. It also gives benchmarks a role beyond ranking models: paired replay can help determine how much historical evidence remains useful, even when benchmark scores transfer poorly to the deployment population. In practice, this motivates retaining versioned patient outcomes and a stable set of replay cases throughout deployment. ECSS makes the benefit of this continuity interpretable by expressing gains in estimation accuracy in units of additional patients.

Limitations and future work. Our experiments use offline replays of graded rewards, with one rollout per model and case. Prospective evaluation on clinical outcomes remains necessary, especially where automated grading has limitations (Appendix B). Hidden updates call for estimators that start at candidate change points, combined with aggregation methods that guarantee small regret on every interval (Daniely et al., 2015; Hazan & Seshadhri, 2009). The evaluated updates are model swaps; prompt, tool and retrieval changes remain to be tested. Another direction is to replace Hedge with parameter-free AdaNormalHedge (Luo & Schapire, 2015) or regret-matching variants (Farina et al., 2023), and evaluate whether these alternatives improve estimation after updates.

AI USE STATEMENT

In this work, we used generative AI tools to assist with methodological development, mathematical derivations and proofs, implementation of methods and experimental pipelines, and analysis of results. Additionally, we used these tools to retrieve and identify related literature, draft sections of the paper, edit and polish prose, and prepare tables and figures from recorded outputs. We did not use generative AI tools to generate synthetic datasets. The evaluated LLMs and benchmark grading procedures are described in Section 5 and Appendix B. The authors directed the study, reviewed the AI-assisted code and text, and verified the theoretical results. We take responsibility for the final content, including all claims and artifacts produced with AI assistance.

ETHICS STATEMENT

This work is a simulation-based study of performance estimation, using offline replay of existing public medical benchmarks to construct simulated deployment sequences. It involves no prospective patient recruitment or interventions in clinical care. The methods and findings are intended for research purposes and do not provide clinical recommendations or replace professional medical judgment. Benchmark scores and automated grades may contain biases and do not establish clinical safety or effectiveness. Any future clinical application would require prospective validation and appropriate clinical oversight.

REPRODUCIBILITY STATEMENT

Our experiments use public medical benchmarks, with model sequences, score pools, and grading procedures documented in Appendix B. Section 5.2 specifies patient sampling, replay cases, and ReMix settings; Appendix D details the estimators, baselines, and their hyperparameters. Metrics and their aggregation, including ECSS matching, are defined in Appendix C. Methods share patient sequences and fixed replay cases across 500 patient replays per target. Appendix E provides the theoretical proofs, and Appendices F and G report ablations and sensitivity analyses.

REFERENCES

Anastasios N. Angelopoulos, Stephen Bates, Clara Fannjiang, Michael I. Jordan, and Tijana Zrnic. Prediction-powered inference. *Science*, 382(6671):669–674, 2023.

Anthropic. Model deprecations. `https://platform.claude.com/docs/en/about-claude/model-deprecations`, 2026a. Accessed September 25, 2026.

Anthropic. Claude platform release notes. `https://platform.claude.com/docs/en/release-notes/overview`, 2026b. Accessed September 25, 2026.

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. HealthBench: Evaluating large language models towards improved human health. arXiv:2505.08775, 2025. URL `https://arxiv.org/abs/2505.08775`.

Apurva Badithela, David Snyder, Lihan Zha, Joseph Mikhail, Matthew O'Kelly, Anushri Dixit, and Anirudha Majumdar. Reliable and scalable robot policy evaluation with imperfect simulators. arXiv:2510.04354, 2025. URL `https://arxiv.org/abs/2510.04354`.

Vishnu Bindu Balachandran. Pay only for disagreement: Certified no-regression verdicts for model updates with matching label-complexity bounds. arXiv:2609.17560, 2026. URL `https://arxiv.org/abs/2609.17560`.

Gagan Bansal, Besmira Nushi, Ece Kamar, Daniel S. Weld, Walter S. Lasecki, and Eric Horvitz. Updates in human-AI teams: Understanding and addressing the performance/compatibility tradeoff. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 2429–2437, 2019.

Suhana Bedi, Yutong Liu, Lucy Orr-Ewing, Dev Dash, Sanmi Koyejo, Alison Callahan, Jason A. Fries, Michael Wornow, Akshay Swaminathan, Lisa Soleymani Lehmann, Hyo Jung Hong, Mehr Kashyap, Akash R. Chaurasia, Nirav R. Shah, Karandeep Singh, Troy Tazbaz, Arnold Milstein, Michael A. Pfeffer, and Nigam H. Shah. Testing and evaluation of health care applications of large language models: A systematic review. *JAMA*, 333(4):319–328, 2025. doi: 10.1001/jama.2024.21700.

Suhana Bedi, Hejie Cui, Miguel Fuentes, Alyssa Unell, Michael Wornow, Juan M. Banda, Nikesh Kotecha, Timothy Keyes, Yifan Mai, Mert Oez, Hao Qiu, Shrey Jain, Leonardo Schettini, Mehr Kashyap, Jason Alan Fries, Akshay Swaminathan, Philip Chung, Fateme Nateghi Haredasht, Ivan Lopez, Asad Aali, Gabriel Tse, Ashwin Nayak, Shivam Vedak, Sneha S. Jain, Birju Patel, Oluseyi Fayanju, Shreya Shah, Ethan Goh, Dong-han Yao, Brian Soetikno, Eduardo Reis, Sergios Gatidis, Vasu Divi, Robson Capasso, Rachna Saralkar, Chia-Chun Chiang, Jenelle Jindal, Tho Pham, Faraz Ghoddusi, Steven Lin, Albert S. Chiou, Hyo Jung Hong, Mohana Roy, Michael F. Gensheimer, Hinesh Patel, Kevin Schulman, Dev Dash, Danton Char, Lance Downing, Francois Grolleau, Kameron Black, Bethel Mieso, Aydin Zahedivash, Wen-wai Yim, Harshita Sharma, Tony Lee, Hannah Kirsch, Jennifer Lee, Nerissa Ambers, Carlene Lugtu, Aditya Sharma, Bilal Mawji, Alex Alekseyev, Vicky Zhou, Vikas Kakkar, Jarrod Helzer, Anurang Revri, Yair Bannett, Roxana Daneshjou, Jonathan Chen, Emily Alsentzer, Keith Morse, Nirmal Ravi, Nima Aghaeepour, Vanessa Kennedy, Akshay Chaudhari, Thomas Wang, Sanmi Koyejo, Matthew P. Lungren, Eric Horvitz, Percy Liang, Michael A. Pfeffer, and Nigam H. Shah. Holistic evaluation of large language models for medical tasks with MedHELM. *Nature Medicine*, 32(3):943–951, 2026. doi: 10.1038/s41591-025-04151-2.

Pierre Boyeau, Anastasios Nikolas Angelopoulos, Tianle Li, Nir Yosef, Jitendra Malik, and Michael I. Jordan. AutoEval done right: Using synthetic data for model evaluation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 5276–5290, 2025. URL https://proceedings.mlr.press/v267/boyeau25a.html.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024. doi: 10.1145/3641289.

Lingjiao Chen, Matei Zaharia, and James Zou. How is ChatGPT's behavior changing over time? *Harvard Data Science Review*, 6(2), 2024. doi: 10.1162/99608f92.5317da47.

Sully F Chen, Anton Alyakin, Andreas Seas, Eunice Yang, Joanne J Choi, Jin Vivian Lee, Amelia L Chen, Pranav I Warman, Rochelle T Bitolas, Robert J Steele, et al. Llm-assisted systematic review of large language models in clinical medicine. *Nature medicine*, 32(3):1152–1159, 2026.

Chenghao Chu and Bingming Yi. Dynamic historical data borrowing using weighted average. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 70(5):1259–1280, 2021. doi: 10.1111/rssc.12512. URL https://academic.oup.com/jrsssc/article/70/5/1259/7033980.

Amit Daniely, Alon Gonen, and Shai Shalev-Shwartz. Strongly adaptive online learning. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1405–1411, 2015.

Jessica Echterhoff, Fartash Faghri, Raviteja Vemulapalli, Ting-Yao Hu, Chun-Liang Li, Oncel Tuzel, and Hadi Pouransari. MUSCLE: A model update strategy for compatible LLM evolution. arXiv:2407.09435, 2024. URL https://arxiv.org/abs/2407.09435.

Gabriele Farina, Julien Grand-Clément, Christian Kroer, Chung-Wei Lee, and Haipeng Luo. Regret matching+: (In)Stability and fast convergence in games. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/c209cd57e13f3344a4cad4ce84d0ee1b-Abstract-Conference.html.

FDA. Considerations for the regulation of generative AI-enabled medical devices: Discussion paper and request for feedback, August 2026. URL https://www.fda.gov/media/194242/download. Docket FDA-2026-N-7874.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Irena Gao, Percy Liang, and Carlos Guestrin. Model equality testing: Which model is this API serving? In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.20247.

Patricia Garcia, Stephen P. Ma, Shreya Shah, Margaret Smith, Yejin Jeong, Anna Devon-Sand, Ming Tai-Seale, Kevin Takazawa, Danyelle Clutter, Kyle Vogt, Carlene Lugtu, Matthew Rojo, Steven Lin, Tait Shanafelt, Michael A. Pfeffer, and Christopher Sharp. Artificial intelligence–generated draft replies to patient inbox messages. *JAMA Network Open*, 7(3):e243201, 2024. doi: 10.1001/jamanetworkopen.2024.3201.

Tarek Haddad, Adam Himes, Laura Thompson, Telba Irony, Rajesh Nair, and MDIC Computer Modeling and Simulation Working Group Participants. Incorporation of stochastic engineering models as prior information in Bayesian medical device trials. *Journal of Biopharmaceutical Statistics*, 27(6):1089–1103, 2017. doi: 10.1080/10543406.2017.1300907.

Elad Hazan and C. Seshadhri. Efficient learning algorithms for changing environments. In *Proceedings of the 26th International Conference on Machine Learning*, pp. 393–400, 2009.

Brian P. Hobbs, Bradley P. Carlin, Sumithra J. Mandrekar, and Daniel J. Sargent. Hierarchical commensurate and power prior models for adaptive incorporation of historical information in clinical trials. *Biometrics*, 67(3):1047–1056, 2011. doi: 10.1111/j.1541-0420.2011.01564.x. URL https://onlinelibrary.wiley.com/doi/full/10.1111/j.1541-0420.2011.01564.x.

Brian P. Hobbs, Daniel J. Sargent, and Bradley P. Carlin. Commensurate priors for incorporating historical information in clinical trials using general and generalized linear models. *Bayesian Analysis*, 7(3):639–674, 2012. doi: 10.1214/12-BA722.

Joseph G. Ibrahim and Ming-Hui Chen. Power prior distributions for regression models. *Statistical Science*, 15(1):46–60, 2000.

Betina Idnay, Zihan Xu, William G Adams, Mohammad Adibuzzaman, Nicholas R Anderson, Neil Bahroos, Douglas S Bell, Cody Bumgardner, Thomas Campion, Mario Castro, et al. Environment scan of generative ai infrastructure for clinical and translational science. *npj Health Systems*, 2 (1):4, 2025.

Sihang Jiang, Lipeng Ma, Zhonghua Hong, Keyi Wang, Zhiyu Lu, Tengfei Wang, Shisong Chen, Jinghao Zhang, Tianjun Pan, Weijia Li, Jiaqing Liang, and Yanghua Xiao. SEA-Eval: A benchmark for evaluating self-evolving agents beyond episodic assessment. arXiv:2604.08988, 2026. URL https://arxiv.org/abs/2604.08988.

Yixing Jiang, Kameron C. Black, Gloria Geng, Danny Park, James Zou, Andrew Y. Ng, and Jonathan H. Chen. MedAgentBench: A virtual EHR environment to benchmark medical LLM agents. *NEJM AI*, 2(9), 2025. doi: 10.1056/AIdbp2500144.

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

Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2Real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics and Automation Letters*, 5(4):6670–6677, 2020. doi: 10.1109/LRA.2020.3013848.

Robert Korom, Sarah Kiptinness, Najib Adan, Kassim Said, Catherine Ithuli, Oliver Rotich, Boniface Kimani, Irene King'ori, Stellah Kamau, Elizabeth Atemba, Muna Aden, Preston Bowman, Michael Sharman, Rebecca Soskin Hicks, Rebecca Distler, Johannes Heidecke, Rahul K. Arora, and Karan Singhal. AI-based clinical decision support for primary care: A real-world study. *arXiv preprint arXiv:2507.16947*, 2025.

Yitao Li. Who drifted: the system or the judge? Anytime-Valid attribution in LLM evaluation pipelines. arXiv:2606.15474, 2026. URL https://arxiv.org/abs/2606.15474.

Cheng Liang, Pengcheng Qiu, Ya Zhang, Yanfeng Wang, Chaoyi Wu, and Weidi Xie. Evaluating large language models in dynamic clinical decision-making with standardized patient cases. arXiv:2606.05112, 2026. URL https://arxiv.org/abs/2606.05112.

Xiaoxuan Liu, Samantha Cruz Rivera, David Moher, Melanie J. Calvert, Alastair K. Denniston, and SPIRIT-AI and CONSORT-AI Working Group. Reporting guidelines for clinical trial reports for interventions involving artificial intelligence: the CONSORT-AI extension. *Nature Medicine*, 26 (9):1364–1374, 2020. doi: 10.1038/s41591-020-1034-x.

Haipeng Luo and Robert E. Schapire. Achieving all with no parameters: AdaNormalHedge. In *Proceedings of the 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pp. 1286–1304. PMLR, 2015. URL https://proceedings.mlr.press/v40/Luo15.html.

Rachel Luo, Heng Yang, Michael Watson, Apoorva Sharma, Sushant Veer, Edward Schmerling, and Marco Pavone. Sim2Val: Leveraging correlation across test platforms for variance-reduced metric estimation. In *Conference on Robot Learning*, 2025. URL https://arxiv.org/abs/2506.20553.

OpenAI. Deprecations, 2026. URL https://developers.openai.com/api/docs/deprecations. Accessed September 25, 2026.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: Evaluating LLMs with fewer examples. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.14992.

R. Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443–1471, 2002. doi: 10.1016/S0378-4266(02)00271-6.

Jacob T. Rosenthal, Ashley Beecy, and Mert R. Sabuncu. Rethinking clinical trials for medical AI with dynamic deployments of adaptive systems. *npj Digital Medicine*, 8:252, 2025. doi: 10.1038/s41746-025-01674-3.

Samuel Schmidgall, Rojin Ziaei, Carl Harris, Eduardo Reis, Jeffrey Jopling, and Michael Moor. AgentClinic: A multimodal agent benchmark to evaluate AI in simulated clinical environments. arXiv:2405.07960, 2024. URL https://arxiv.org/abs/2405.07960.

Heinz Schmidli, Sandro Gsteiger, Satrajit Roychoudhury, Anthony O'Hagan, David Spiegelhalter, and Beat Neuenschwander. Robust meta-analytic-predictive priors in clinical trials with historical control information. *Biometrics*, 70(4):1023–1032, 2014.

Yueqi Shen, Matthew A. Psioda, and Joseph G. Ibrahim. BayesPPD: An R package for bayesian sample size determination using the power and normalized power prior for generalized linear models. *The R Journal*, 14(4):335–351, 2022. doi: 10.32614/RJ-2023-016.

Jeremy Smith and Kenneth F. Wallis. A simple explanation of the forecast combination puzzle. *Oxford Bulletin of Economics and Statistics*, 71(3):331–355, 2009.

Megha Srivastava, Besmira Nushi, Ece Kamar, Shital Shah, and Eric Horvitz. An empirical analysis of backward compatibility in machine learning systems. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 3272–3280, 2020.

James H. Stock and Mark W. Watson. Combination forecasts of output growth in a seven-country data set. *Journal of Forecasting*, 23(6):405–430, 2004.

Aaron A. Tierney, Gregg Gayre, Brian Hoberman, Britt Mattern, Manuel Ballesca, Patricia Kipnis, Vincent Liu, and Kristine Lee. Ambient artificial intelligence scribes to alleviate the burden of clinical documentation. *NEJM Catalyst*, 5(3), 2024. doi: 10.1056/CAT.23.0404.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. Anchor points: Benchmarking models with much fewer examples. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1576–1601, 2024. doi: 10.18653/v1/2024.eacl-long.95.

Vladimir Vovk. Competitive on-line statistics. *International Statistical Review*, 69(2):213–248, 2001.

Manuel Wiesenfarth and Silvia Calderazzo. Quantification of prior impact in terms of effective current sample size. *Biometrics*, 76(1):326–336, 2020. doi: 10.1111/biom.13124.

Skyler Wu, Yash Nair, and Emmanuel J. Candès. Efficient evaluation of LLM performance with statistical guarantees. arXiv:2601.20251, 2026. URL https://arxiv.org/abs/2601.20251.

Sijie Yan, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqi Deng, Meng Wang, Wei Xia, and Stefano Soatto. Positive-congruent training: Towards regression-free model updates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14299–14308, 2021.

Peng Yang, Yuansong Zhao, Lei Nie, Jonathon Vallejo, and Ying Yuan. SAM: Self-adapting mixture prior to dynamically borrow information from historical data in clinical trials. *Biometrics*, 79(4): 2857–2868, 2023. doi: 10.1111/biom.13927.

Guangtao Zeng, Wenmian Yang, Zeqian Ju, Yue Yang, Sicheng Wang, Ruisi Zhang, Meng Zhou, Jiaqi Zeng, Xiangyu Dong, Ruoyu Zhang, Hongchao Fang, Penghui Zhu, Shu Chen, and Pengtao Xie. MedDialog: Large-scale medical dialogue datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9241–9250, 2020. doi: 10.18653/v1/2020.emnlp-main.743.

Wen Zhang, Zhiying Pan, and Ying Yuan. A bayesian group sequential design for randomized biosimilar clinical trials with adaptive information borrowing from historical data. *Journal of Biopharmaceutical Statistics*, 32(3):359–372, 2022. doi: 10.1080/10543406.2022.2080700.

Lina Zhao, Shengrui Liu, Tangsiwei Xin, Jiawen Tan, Xiaoran Wang, Yafang Li, Zihao Bian, Yiyang Chen, Fanyi Kong, Jinwei Bian, Chen Qian, and Zongjiu Zhang. AI agent in healthcare: applications, evaluations, and future directions. *npj Artificial Intelligence*, 2:31, 2026. doi: 10.1038/s44387-026-00076-4.

Congjie Zheng, Chuanyi Xue, Bin Liang, Jun Yang, and Changshui Zhang. SEAGym: An evaluation environment for self-evolving LLM agents. arXiv:2606.17546, 2026. URL https://arxiv.org/abs/2606.17546.

A NOTATION

Table 2: Main notation.

Symbol	Meaning
Evaluation stream	
$t,\ T$	Global case index and total number of observed cases.
$k,\ K,\ \pi_k$	Version index, number of versions, and deployed system version k.
t_k	Onset time of version k (its first case); $t_1 = 1$ and $t_{K+1} = T + 1$.
$k(t)$	Version serving case t: $k(t) = k$ for $t_k \le t < t_{k+1}$.
$h,\ L_k$	Within-version case index $h = t - t_k + 1$ and version length $L_k = t_{k+1} - t_k$.
$\mathcal{P}$	Fixed target population.
$r_t,\ r_h^{(k)}$	Observed reward, indexed globally or within a version.
$v^{(k)},\ v_t$	Mean reward of version k; $v_t = v^{(k(t))}$.
δ_k	Change in mean reward: $v^{(k)} - v^{(k-1)}$.
$\hat{v}_t,\ \hat{v}_{k,h}$	After-case performance estimate; $\hat{v}_{k,h} = \hat{v}_{t_k+h-1}$.
$\mathcal{F}_{k,h}$	Information available after h patients of version k, including history and available replay evaluations.
$\mathcal{E}_T$	Cumulative squared estimation error, $\sum_{t=1}^{T}(\hat{v}_t - v_t)^2$.
Benchmark replay	
$m,\ M$	Benchmark index and number of benchmarks.
$j,\ J_m$	Replay-case index and number of selected cases from benchmark m; $J_m = 20$ in the main experiments.
$\tilde{r}_j^{(m,k)}$	Reward of version k on anchor case j of benchmark m.
$d_k^{(m)}$	Mean absolute paired reward difference between versions $k-1$ and k on its J_m fixed simulated cases.
Borrowing and aggregation	
$i,\ N$	Estimator index and number of constituent estimators.
$\tilde{\mathbf{v}}_t,\ \tilde{\mathbf{v}}_{k,h}$	Vector of estimator outputs after the current case; component i is $\tilde{v}_t(i)$.
$\mathbf{w}_t,\ \mathbf{w}_{k,h}$	Vector of normalized aggregation weights; component i is $w_t(i)$.
$\mathbf{1}$	All-ones vector; vector squares and exponentials act componentwise.
$\boldsymbol{\mu}_k,\ \mathbf{P}_k$	Carried estimator means and working variances; borrowing estimator i uses $\mu_k(i), P_k(i)$.
$\bar{r}_h^{(k)},\ \hat{\sigma}_{k,h}^2$	Current version's sample mean and plug-in reward variance.
n_0	Effective pseudo-count in the borrowing analysis.
$\lambda_{k,h}(i)$	Weight on the carried estimate within a Bayesian borrowing estimator.
η	Learning rate for squared-loss exponential weighting.
$\boldsymbol{\ell}_t,\ \hat{\ell}_t$	Estimator loss vector and ReMix's loss; both are zero at version onsets.
$\mathcal{R}_T$	Cumulative regret against the best fixed estimator through patient T.
ϵ	Replay mismatch and sampling-error allowance in Appendix E.4.
$\mathcal{E}_T^{\mathrm{curr}},\ G_T(i)$	Current-only cumulative estimation error and estimator i's expected cumulative improvement over it.
Evaluation metrics	
$\mathrm{MSE}(t)$	Expected squared estimation error, $\mathbb{E}[(\hat{v}_t - v_t)^2]$.
$\mathrm{CVaR}_\alpha(X)$	Mean of the worst $1-\alpha$ fraction of loss X; experiments use $\alpha = 0.95$.
$Q_a(h),\ B_h(n)$	Relative MSE of method a at h, averaged across targets, and the identically normalized current-only reference curve.
$n_a^\star(h),\ \mathrm{ECSS}_a(h)$	Matched total current sample size and additional current sample size for method a, $n_a^\star(h) - h$.

B DATA AND REPLAY PROTOCOL

B.1 MEDHELM

The targets are MedDialog-HealthCareMagic, MedDialog-iCliniq, and MedicationQA. The first two contain patient questions from online consultation platforms; the third contains consumer medica-

tion questions. We use per-instance scores from MedHELM v4.0.0 (Bedi et al., 2026), mapping jury scores on a 1–5 scale to $[0, 1]$.

The chronological sequence is GPT-4o, Gemini-1.5-Pro, GPT-4o-mini, Claude-3.5-Sonnet, Llama-3.3-70B, DeepSeek-R1, o3-mini, Gemini-2.0-Flash, Claude-3.7-Sonnet, o4-mini, Gemini-2.5-Pro, GPT-5, and GPT-5-mini. The closed-model subset omits Llama-3.3-70B and DeepSeek-R1. Both sequences use 50 patients per version and the same target pools: 1,000 HealthCareMagic cases, 1,000 iCliniq cases, and 689 MedicationQA cases. Model order follows recorded checkpoint release dates, with full model identifiers breaking same-day ties.

The four replay sources are ACI-Bench, HeadQA, MedBullets, and MedCalc-Bench, with respectively 120, 1,000, 308, and 1,000 jointly scored cases. They are the first four eligible external sources alphabetically by benchmark name, with eligibility requiring at least 20 common cases across all versions. We sample 20 cases per source once for each target and sequence, then fix their identities and stored scores across all 500 patient replays.

B.2 Supplementary Testbed

HealthBench provides health conversations and response-specific grading rubrics (Arora et al., 2025); MedAgentBench provides EHR tool-use tasks with a rule-based checker (Jiang et al., 2025); and MedSP provides simulated-patient interactions (Liang et al., 2026). Each benchmark serves as a target, with the other two supplying external replay cases.

The chronological sequence is Llama-3-8B-Instruct, Med42-v2-8B, Gemma-3-4B-IT, MedGemma-4B-IT, GPT-5.4-mini, and GPT-5.6-luna. The open-model subset uses the first four. Both sequences use 30 patients per version and the same six-model paired pools: 897 HealthBench cases, 360 MedAgentBench cases, and 410 MedSP cases. Each external source supplies 20 fixed cases per target and sequence.

HealthBench responses were graded by GPT-4.1. MedSP uses DeepSeek-V4-Pro for its auxiliary patient, environment-controller, and evaluator roles. MedAgentBench uses its task checker and produces binary rewards. HealthBench and MedSP rewards are graded scores in $[0, 1]$.

B.3 Shared offline replay protocol

Scored pools. Cases must have usable scores for every included version; low scores alone are not grounds for exclusion. Failed doctor-side MedSP encounters and unscored MedAgentBench checker exceptions are excluded when constructing the paired pools. Scored behavioral failures are retained. MedAgentBench retains the checker's reported rewards, including no-op passes and passes from some nonterminating dialogues. The estimand is this graded reward.

Patient sampling and benchmark replay. Each target and sequence has 500 independent patient orderings, shared by all methods. We randomly permute the paired target pool, reserve the first 20 positions, and partition the next KL positions into K successive version blocks of length L. Thus each patient case appears at most once in a stream. Fixed replay-source cases are drawn independently of patient orderings; stored outcomes are reused without new generation or grading. The four sequence settings, each with three targets, give 6,000 patient replays in total. Table 1 reports the two full sequences; Appendix F reports the two model subsets and estimator ablations from this same experiment.

C Metric definitions and aggregation

For each fixed position t, we estimate conditional MSE from the 500 patient replays as

$$\widehat{\mathrm{MSE}}(t) = \frac{1}{500} \sum_{b=1}^{500} (\hat{v}_t^{(b)} - v_t)^2, \tag{11}$$

where b indexes patient replays and all methods share the sampled cases. These empirical metrics condition on the model sequence, update times, replay anchors, and stored scores; only patient sampling varies. The target v_t is the deployed version's full-pool mean.

Averaging updates. Let a index methods, with $a = 0$ for current-only, and let s index the three target streams in each testbed. Write $\mathbb{E}_s$ for expectation over patient replays. Define

$$\begin{aligned} \mathrm{MSE}_{a,s}(h) &= \frac{1}{K_s - 1} \sum_{k=2}^{K_s} \mathbb{E}_s[(\hat{v}^a_{s,k,h} - v^{(s,k)})^2], \\ C^{5\%}_{a,s}(h) &= \frac{1}{K_s - 1} \sum_{k=2}^{K_s} \mathrm{CVaR}_{0.95,s}((\hat{v}^a_{s,k,h} - v^{(s,k)})^2). \end{aligned} \tag{12}$$

Thus @ h refers to the hth observation of every updated version, excluding the initial version. Each testbed has $S = 3$ streams.

Relative summaries. For one testbed, define

$$Q_a(h) = \frac{1}{S} \sum_{s=1}^{S} \frac{\mathrm{MSE}_{a,s}(h)}{\mathrm{MSE}_{0,s}(h)}, \qquad Q^{5\%}_a(h) = \frac{1}{S} \sum_{s=1}^{S} \frac{C^{5\%}_{a,s}(h)}{C^{5\%}_{0,s}(h)}. \tag{13}$$

These are the relative MSE@ h and CVaR@ h entries in the result tables. Normalization precedes averaging across targets. All reported current-only denominators are positive.

Tail risk. At each (s, k, h), empirical CVaR is the mean of the 25 largest squared errors among 500 replays. Equation (12) then averages these tails across updates.

Sample efficiency. Adapting the MSE-matching principle of Wiesenfarth & Calderazzo (2020), we define the current-only reference curve

$$B_h(n) = \frac{1}{S} \sum_{s=1}^{S} \frac{\mathrm{MSE}_{0,s}(n)}{\mathrm{MSE}_{0,s}(h)}. \tag{14}$$

The denominator stays fixed at h as the candidate current sample size n varies, so $B_h(h) = 1$. We use the same updated versions and targets at every candidate n.

For the empirical calculation, substitute the replay averages from Equation (11) to obtain $\widehat{B}_h(n)$ and $\widehat{Q}_a(h)$. Fit $\widehat{B}_h$ by nonincreasing least-squares isotonic regression subject to the anchor constraint $\widetilde{B}_h(h) = 1$, then linearly interpolate between integer sample sizes. With $n_{\max}$ the common supported horizon, the reported sample-size match is

$$\begin{aligned} \mathcal{M}_a(h) &= \{n \in [1, n_{\max}] : \widetilde{B}_h(n) = \widehat{Q}_a(h)\}, \\ n^\star_a(h) &= \operatorname*{arg\,min}_{n \in \mathcal{M}_a(h)} |n - h|, \qquad \mathrm{ECSS}_a(h) = n^\star_a(h) - h. \end{aligned} \tag{15}$$

The closest match to h resolves plateaus and ensures current-only has ECSS zero. Inversion follows averaging across patient replays, updates, and normalized targets; we do not average separately inverted sample sizes. The supported range is 1–50 patients for MedHELM and 1–30 for the Supplementary Testbed. If the method's risk is below the fitted curve at $n_{\max}$, we report $\mathrm{ECSS}_a(h) > n_{\max} - h$; if it is above the curve at 1, we report $\mathrm{ECSS}_a(h) < 1 - h$. We do not extrapolate beyond the supported range.

Total MSE and Total CVaR. Relative Total MSE is $S^{-1} \sum_s \mathbb{E}[\mathcal{E}^a_{T_s}] / \mathbb{E}[\mathcal{E}^0_{T_s}]$, using Equation (2) and including the initial version. Total CVaR is

$$\frac{1}{S} \sum_{s=1}^{S} \frac{\mathrm{CVaR}_{0.95,s}(\mathcal{E}^a_{T_s})}{\mathrm{CVaR}_{0.95,s}(\mathcal{E}^0_{T_s})}. \tag{16}$$

Here CVaR averages the 25 largest cumulative errors among the 500 replays for each target; it is not a sum of position-wise CVaRs. All reported values are point estimates.

D Estimator implementation

D.1 Borrowing states and variance estimates

Let $s^2_{k,h}$ be the sample reward variance for $h \geq 2$. Initialize the fallback variance $f_2 = \max\{10^{-3}, s^2_{1,L_1}\}$. Within version $k \geq 2$, set $\hat{\sigma}^2_{k,h} = \max\{10^{-3}, s^2_{k,h}\}$ when $h \geq 2$ and $s^2_{k,h} > 0$,

and use f_k otherwise. After the version, set $f_{k+1} = \max\{10^{-3}, s^2_{k,L_k}\}$ if $s^2_{k,L_k} > 0$, and retain f_k otherwise.

For each dissimilarity-based estimator, initialize $\mu_2(i) = \bar{r}^{(1)}_{L_1}$ and $P_2(i) = f_2/L_1$. Each estimator then carries its own final mean and working variance:

$$\mu_{k+1}(i) = \tilde{v}_{k,L_k}(i), \qquad P_{k+1}(i) = (1 - \lambda_{k,L_k}(i))\frac{\hat{\sigma}^2_{k,L_k}}{L_k}.$$

D.2 Comparison estimators

Initialization and history. All methods use the current empirical mean throughout the initial version. Thereafter, pooling uses all observed patient outcomes and the sliding window uses the most recent 40, spanning version boundaries. The weighted-average estimator of Chu & Yi (2021) uses only the immediately preceding version as history. The five Bayesian methods use all completed versions as historical studies; CP assumes a common historical mean. These six methods also form the prior-work components of ReMix.

Weighted-average borrowing. The estimator of Chu & Yi (2021) combines the preceding version's mean $\bar{r}^{(k-1)}_{L_{k-1}}$ and the current mean $\bar{r}^{(k)}_h$. Let u_{hist} and u_{curr} denote estimated variances of these means. The historical weight is

$$w_{\text{hist}} = \frac{u_{\text{curr}}}{u_{\text{curr}} + u_{\text{hist}} + (\bar{r}^{(k)}_h - \bar{r}^{(k-1)}_{L_{k-1}})^2}.$$

Each mean's variance is its sample reward variance divided by its sample size. At $h = 1$, the current reward variance is taken from the preceding version. A zero denominator gives zero historical weight.

Likelihoods. Binary endpoints use Bernoulli likelihoods. For graded rewards, rMAP, SAM, EMAP, and CP use a normal working likelihood with observation standard deviation 0.5; NPP integrates an unknown shared precision.

MAP mixtures. rMAP assigns equal prior mass to a hierarchical MAP component and a weak component. SAM uses the same components with a likelihood-ratio weight and discrepancy parameter 0.10. For these two methods, the hierarchical mean has prior $\mathcal{N}(0, 2^2)$ on the logit scale for binary rewards and $\mathcal{N}(0.5, 1)$ for graded rewards. The between-study standard deviation has a half-normal prior with scale 1 and 0.25, respectively. The weak component is $\text{Beta}(1, 1)$ for binary rewards and a normal prior centered at the MAP predictive mean with variance 0.5^2 for graded rewards.

Elastic and other adaptive priors. EMAP uses the published step cutoff 0.4384 from the equal-allocation example in Zhang et al. (2022, Table A4). The graded-reward version uses the same cutoff and a normal hierarchical working model. CP uses a Gamma(shape 1, rate 1) commensurability precision. NPP jointly normalizes the powered historical likelihoods, with independent uniform powers on $[0, 1]$ for binary rewards and $[0.05, 1]$ for graded rewards (Shen et al., 2022). Its initial prior is $\text{Beta}(1, 1)$ for binary rewards; for graded rewards, it is flat in the mean with precision density proportional to the inverse precision.

E Proofs

E.1 Paired dissimilarity

Fix update $k \geq 2$ and let $\delta_k = v^{(k)} - v^{(k-1)}$. Write $\Delta_j = \tilde{r}^{(m,k)}_j - \tilde{r}^{(m,k-1)}_j$. Let $\Delta(x) = \mathbb{E}[r^{(k)}(x) - r^{(k-1)}(x)]$ be the expected effect of the update on case x, where the expectation is over rollouts and grading.

Proposition 2 (Upper bound on the performance change under covariate shift)**.** *Suppose benchmark m and the real population share a case space and a reward function, benchmark cases are drawn from Q_m, and $W_m = \|d\mathcal{P}/dQ_m\|_\infty < \infty$. Then*

$$|\delta_k| \;\leq\; \mathbb{E}_{\mathcal{P}}|\Delta(x)| \;\leq\; W_m\, \mathbb{E}_{Q_m}|\Delta(x)| \;\leq\; W_m\, \mathbb{E}\big[d^{(m)}_k\big].$$

When the benchmark follows the real population, $W_m = 1$ and $\mathbb{E}[d_k^{(m)}] \geq |\delta_k|$.

Proof of Proposition 2. Because the real population and the benchmark share the reward function, $v^{(k)} = \mathbb{E}_{\mathcal{P}}\mathbb{E}[r^{(k)}(x)]$ and $\delta_k = \mathbb{E}_{\mathcal{P}}\Delta(x)$. Jensen's inequality gives $|\delta_k| \leq \mathbb{E}_{\mathcal{P}}|\Delta(x)|$. Changing measure,

$$\mathbb{E}_{\mathcal{P}}|\Delta(x)| = \mathbb{E}_{Q_m}\Big[\frac{d\mathcal{P}}{dQ_m}(x)\,|\Delta(x)|\Big] \leq W_m\,\mathbb{E}_{Q_m}|\Delta(x)|.$$

Finally, for an anchor case $x_j \sim Q_m$, Jensen's inequality over rollouts and grading gives $|\Delta(x_j)| \leq \mathbb{E}\big[|\Delta_j| \,\big|\, x_j\big]$, so $\mathbb{E}_{Q_m}|\Delta(x)| \leq \mathbb{E}|\Delta_j| = \mathbb{E}[d_k^{(m)}]$. □

E.2 Squared-loss aggregation

Squared loss on $[0, 1]$ is 2-mixable, and Equation (6) is its standard substitution (Vovk, 2001; Cesa-Bianchi & Lugosi, 2006).

Proof of Theorem 1. Let $Z_0 = 1$ and $Z_t = N^{-1}\sum_i \exp(-\eta\sum_{s=1}^{t}\ell_s(i))$. At every non-onset time, mixability gives

$$\hat{\ell}_t \leq -\frac{1}{\eta}\log\sum_i w_{t-1}(i)e^{-\eta\ell_t(i)} = -\frac{1}{\eta}\log\frac{Z_t}{Z_{t-1}}.$$

At an onset, all losses are zero and $Z_t = Z_{t-1}$, so the same inequality holds. Summing and using $Z_T \geq N^{-1}\exp(-\eta\min_i\sum_t\ell_t(i))$ yields Equation (7). □

E.3 After-case estimation and the boundary term

Suppose $\mathbb{E}[r_h^{(k)} \mid \mathcal{F}_{k,h-1}] = v^{(k)}$, where the past includes available anchors and all estimator states. For any predictable a,

$$\mathbb{E}[(r_h^{(k)} - a)^2 \mid \mathcal{F}_{k,h-1}] = \mathrm{Var}(r_h^{(k)} \mid \mathcal{F}_{k,h-1}) + (v^{(k)} - a)^2. \tag{17}$$

Define $\mathcal{E}_T^i = \sum_k\sum_{h=1}^{L_k}(\tilde{v}_{k,h}(i) - v^{(k)})^2$.

Corollary 2 (After-case estimation)**.** *Under the conditional-mean assumption,*

$$\mathbb{E}\mathcal{E}_T \leq \min_{i\in[N]}\mathbb{E}\mathcal{E}_T^i + \frac{\log N}{\eta} + K. \tag{18}$$

The boundary term K accounts for each version's last after-case estimate.

Proof of Corollary 2. Fix an estimator index i. Apply Equation (17) to ReMix and this estimator at each non-onset time. The conditional reward variance cancels. Taking expectations in Equation (7) gives

$$\mathbb{E}\sum_k\sum_{h=1}^{L_k-1}(\hat{v}_{k,h} - v^{(k)})^2 \leq \mathbb{E}\sum_k\sum_{h=1}^{L_k-1}(\tilde{v}_{k,h}(i) - v^{(k)})^2 + \frac{\log N}{\eta}.$$

Each after-case estimate and target is in $[0, 1]$, so the learner's omitted terminal loss on each version is at most one. Add these losses, and then add the comparator's nonnegative terminal losses to the right-hand side. This proves the full after-case bound with K. Taking the minimum of the expected risks over fixed estimator indices completes the argument.

□

With sampling without replacement, the next reward's conditional mean is the remaining-pool mean. The full-pool empirical target therefore need not satisfy this corollary's conditional-mean assumption; Theorem 1 holds for either sampling scheme.

E.4 From replay accuracy to mixture accuracy

Replay accuracy. Using the shared case space and reward function of Appendix E.1, let $\Delta(x)$ be the expected reward change and Q the replay distribution. If

$$|\mathbb{E}_{\mathcal{P}}\Delta - \mathbb{E}_{Q}\Delta| \le \epsilon_{\text{pop}}, \qquad \mathbb{E}_{Q}|\Delta| \le d + \epsilon_{\text{sample}},$$

then $|\delta_k| \le d + \epsilon$ with $\epsilon = \epsilon_{\text{pop}} + \epsilon_{\text{sample}}$. Here ϵ_{pop} measures transfer error in the update's effect, and ϵ_{sample} accounts for finite replay samples and rollout noise.

Proof of Proposition 1. Condition on $\mathcal{F}_{k,0}$, fixing μ, P, and d. By the triangle inequality and $(a+b)^2 \le 2(a^2+b^2)$,

$$(\mu - v^{(k)})^2 \le (\sqrt{P} + d)^2 \le 2(P + d^2) = 2q.$$

With the true variance, Equation (4) is shrinkage with pseudo-count $n_0 = \sigma^2/q$. The conditional MSE is

$$\frac{h\sigma^2 + n_0^2(\mu - v^{(k)})^2}{(h+n_0)^2} \le \frac{(h+2n_0)\sigma^2}{(h+n_0)^2} = \frac{\sigma^2}{h} - \frac{\sigma^2 n_0^2}{h(h+n_0)^2}.$$

The reduction is strictly positive for $h \ge 1$. If $q = 0$ and the calibration assumptions hold, the carried mean equals the target exactly and the borrowing estimator has zero error. □

Calibration and replay error. Proposition 1 assumes the true reward variance, a bound on historical squared error, and a bound on the update's performance change. In the implementation, $\hat{\sigma}^2_{k,h}$ is a plug-in variance, $P_k(i)$ is a working variance, and $d_k^{(m)}$ is a finite-sample replay statistic. The guarantee applies when these quantities satisfy the stated calibration conditions. With the true reward variance and $|\delta_k| \le d + \epsilon$, the same proof applies whenever

$$(|\mu - v^{(k-1)}| + d + \epsilon)^2 \le 2(P + d^2).$$

Proof of Corollary 1. For a fixed estimator rule i, write $G_T(i) = \mathbb{E}\mathcal{E}_T^{\text{curr}} - \mathbb{E}\mathcal{E}_T^i$. Corollary 2 gives $\mathbb{E}\mathcal{E}_T \le \mathbb{E}\mathcal{E}_T^i + \log N/\eta + K$. Substituting the definition of $G_T(i)$ proves Equation (9).

For example, suppose the same estimator rule uses current-only estimation on the initial version and satisfies Proposition 1 almost surely on every later version, with version-specific variance σ_k^2 and pseudo-count $n_{0,k}$. Summing its conditional MSE gains and taking expectations yields

$$G_T(i) \ge \sum_{k=2}^{K}\sum_{h=1}^{L_k} \mathbb{E}\left[\frac{\sigma_k^2 n_{0,k}^2}{h(h+n_{0,k})^2}\right].$$

□

F Estimator ablations and model subsets

Estimator ablations. Table 3 compares five collections under the protocol in Section 5.2. Prior-work only combines the six published borrowing methods; replay only combines the two estimators per source. The two intermediate collections add either dissimilarity-based borrowing or benchmark-score estimates to the six prior-work methods. Every collection uses Algorithm 1 with $\eta = 2$.

On the full MedHELM sequence, the combined collection improves relative Total MSE from 0.797 for prior-work only and 0.761 for replay only to 0.706. On the full Supplementary Testbed sequence, replay only performs best, with relative Total MSE 0.619 versus 0.662 for ReMix. Adding both replay types to prior-work methods improves mean accuracy over adding either type alone on both full sequences.

Model subsets. Table 4 reports all standalone comparisons for the 11-version MedHELM closed-model sequence and the four-version Supplementary Testbed open-model sequence. ReMix has the lowest mean error in the former, while NPP has lower mean error in the latter. NPP's Total MSE advantage on the open-model subset is driven by MedSP, whose mean rewards change relatively

Table 3: **Estimator ablations.** Risks are relative to current-only; ECSS counts additional equivalent patients. Collections share patient replays and anchors within each sequence.

	After 5 patients			Total	
Method	ECSS ↑	MSE ↓	CVaR ↓	MSE ↓	CVaR ↓
MedHELM: full sequence					
Prior-work only	+2.51	0.663	0.461	0.797	0.655
Replay only	+2.01	0.708	0.490	0.761	0.587
Prior-work + dissimilarity	+3.08	0.614	0.433	0.711	0.579
Prior-work + benchmark scores	+2.62	0.653	0.467	0.794	0.651
ReMix	+3.14	0.610	0.439	0.706	0.575
MedHELM: closed-model subset					
Prior-work only	+2.11	0.711	0.490	0.860	0.697
Replay only	+2.30	0.693	0.451	0.766	0.621
Prior-work + dissimilarity	+2.67	0.659	0.460	0.766	0.638
Prior-work + benchmark scores	+2.86	0.641	0.448	0.798	0.654
ReMix	+3.17	0.616	0.431	0.724	0.609
Supplementary Testbed: full sequence					
Prior-work only	+1.42	0.781	0.680	0.747	0.779
Replay only	+3.45	0.582	0.478	0.619	0.666
Prior-work + dissimilarity	+1.92	0.719	0.651	0.700	0.734
Prior-work + benchmark scores	+2.19	0.690	0.581	0.696	0.728
ReMix	+2.58	0.652	0.574	0.662	0.697
Supplementary Testbed: open-model subset					
Prior-work only	+5.29	0.485	0.533	0.591	0.644
Replay only	+8.35	0.372	0.302	0.531	0.591
Prior-work + dissimilarity	+5.24	0.488	0.515	0.587	0.640
Prior-work + benchmark scores	+7.64	0.393	0.406	0.536	0.602
ReMix	+7.13	0.406	0.420	0.539	0.605

little across versions (0.255, 0.254, 0.287, and 0.211). This pattern is consistent with historical borrowing being more beneficial when performance remains relatively stable, although it does not isolate version-to-version change as the cause. These subsets retain the same patient counts per version, target pools, and source-selection rule as the full sequences.

Table 4: **Model-subset results.** Reporting and normalization follow Table 1; each subset averages three target ratios from 500 patient replays per target.

	After 5 patients			Total	
Method	ECSS ↑	MSE ↓	CVaR ↓	MSE ↓	CVaR ↓
MedHELM: closed-model subset					
Pooling	-2.10	1.725	0.379	2.930	1.544
Sliding window (40)	-2.39	1.961	0.568	1.625	1.007
Chu & Yi (2021)	+1.61	0.768	0.649	0.827	0.802
rMAP	+1.08	0.832	0.418	1.237	0.816
SAM	+1.16	0.822	0.435	1.134	0.783
EMAP	-1.09	1.293	0.668	1.490	1.167
CP	+0.74	0.883	0.884	0.858	0.816
NPP	-1.16	1.321	0.359	1.715	1.033
ReMix	+3.17	0.616	0.431	0.724	0.609
Supplementary Testbed: open-model subset					
Pooling	+6.28	0.439	0.218	0.644	0.665
Sliding window (40)	+6.30	0.438	0.259	0.586	0.645
Chu & Yi (2021)	+1.75	0.737	0.762	0.777	0.812
rMAP	+6.27	0.439	0.478	0.557	0.621
SAM	+5.44	0.478	0.531	0.582	0.643
EMAP	+0.57	0.906	0.992	0.943	0.971
CP	+1.88	0.721	0.740	0.749	0.778
NPP	+8.46	0.368	0.221	0.530	0.596
ReMix	+7.13	0.406	0.420	0.539	0.605

G SENSITIVITY TO VERSION LENGTH

Setup. We vary the number of patients per version, $L \in \{25, 30, 35, 40, 50\}$, on the same full chronological sequences as Table 1. Every setting retains all three targets in each suite, the same model order, the same 20 fixed anchors per source, and the same estimator configurations. Each target uses 500 patient replays. Within a replay, shorter versions use prefixes of the longer versions' patient draws, and no patient is observed twice in a stream. The default settings, $L = 50$ for MedHELM and $L = 30$ for the Supplementary Testbed, reproduce the main-table predictions exactly. All other lengths recompute the estimators on the resulting histories. The sliding window remains fixed at 40 patients; EMAP's planned version length follows L.

The shortest length preserves the graded NPP configuration used in the main comparison. Its power lower bound of 0.05 requires the weighted historical count to exceed one, hence $0.05L > 1$ at the first update. We therefore do not include lengths 10 or 20, which would require changing this configuration. The model-version count is fixed, so total patient budget increases with L; this experiment does not isolate update frequency at a fixed total budget.

Results. ReMix has the lowest mean whole-stream MSE among the compared methods at every tested length in both suites (Table 5). Its relative MSE increases from 0.609 at $L = 25$ to 0.706 at $L = 50$ on MedHELM, and from 0.646 to 0.706 on the Supplementary Testbed. Thus its advantage over current-only is larger when each version serves fewer patients, while remaining present throughout the tested range (Figure 4). Early post-update accuracy is comparatively stable: ReMix's relative MSE after five patients ranges from 0.606 to 0.610 on MedHELM and from 0.652 to 0.655 on the Supplementary Testbed. These comparisons describe conditional mean risks, without a claim of statistical significance across lengths.

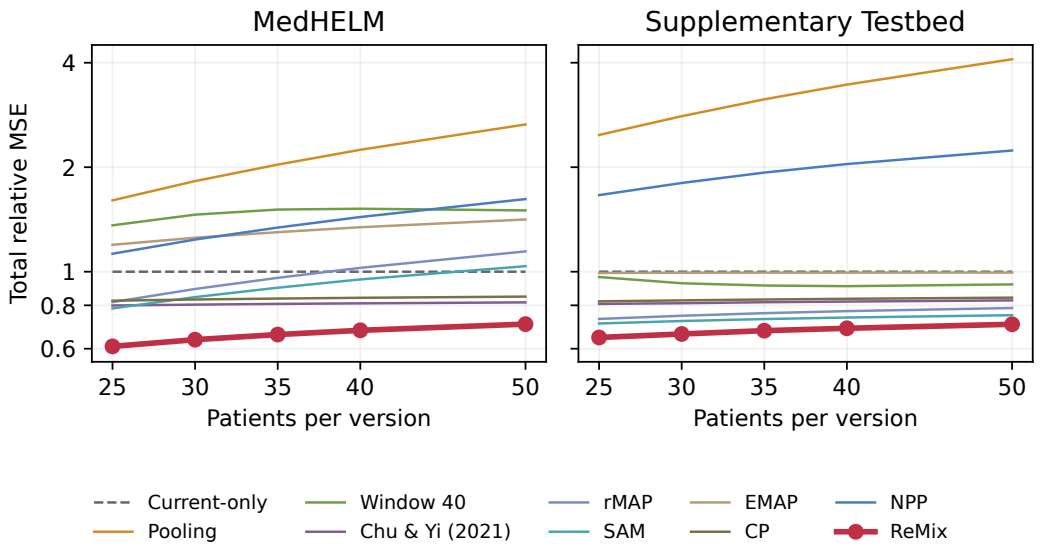

Figure 4: **Sensitivity to the number of patients per version.** Whole-stream squared error is averaged over 500 patient replays, normalized by current-only within each target, and then averaged over the same three targets at every length. Both suites use their full chronological model sequences and fixed source anchors. Curves show point estimates conditional on the stored score pools; the vertical axis is logarithmic.

Table 5: **Whole-stream relative MSE across version lengths.** Every column retains all targets and methods. Values below one improve on current-only; bold marks the lowest mean in each suite and column. Initial-version error is included.

Method	$L = 25$	$L = 30$	$L = 35$	$L = 40$	$L = 50$
MedHELM					
Current-only	1.000	1.000	1.000	1.000	1.000
Pooling	1.605	1.824	2.035	2.245	2.656
Window 40	1.360	1.460	1.510	1.518	1.502
Chu & Yi (2021)	0.799	0.804	0.808	0.811	0.816
rMAP	0.817	0.892	0.960	1.026	1.144
SAM	0.784	0.845	0.899	0.950	1.037
EMAP	1.195	1.253	1.300	1.343	1.414
CP	0.825	0.832	0.837	0.841	0.848
NPP	1.127	1.238	1.340	1.437	1.619
ReMix	**0.609**	**0.637**	**0.659**	**0.678**	**0.706**
Supplementary Testbed					
Current-only	1.000	1.000	1.000	1.000	1.000
Pooling	2.477	2.805	3.138	3.460	4.094
Window 40	0.965	0.926	0.912	0.908	0.919
Chu & Yi (2021)	0.808	0.811	0.817	0.820	0.827
rMAP	0.731	0.747	0.760	0.770	0.786
SAM	0.709	0.721	0.730	0.738	0.749
EMAP	0.991	0.994	0.994	0.994	0.994
CP	0.821	0.827	0.832	0.835	0.842
NPP	1.662	1.801	1.930	2.042	2.234
ReMix	**0.646**	**0.662**	**0.676**	**0.687**	**0.706**