SURVIVAL WORLD: EXPLORING ALTERNATIVE PATIENT FUTURES THROUGH TRAJECTORY REASONING FOR SURVIVAL PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Longitudinal survival prediction requires reasoning about how a patient's condition may evolve beyond the observed history. Existing survival prediction methods focus on estimating risk scores or survival probabilities, without explicitly representing the alternative disease trajectories that may unfold beyond the observed history. We introduce SURVIVAL WORLD (SW), a programmable empirical world model that predicts survival by assembling alternative future trajectories from observed patient transitions. A large language model translates feature descriptions and aggregate evaluation feedback into a bounded retrieval program that identifies relevant transitions at each visited state. Executing this program generates multiple rollout paths, which shared survival predictors evaluate and combine into a survival forecast. Each step retains its supporting observed transition, allowing the forecast to be traced to empirical evidence and alternative patient futures to be compared. Experiments on PBCseq and a second-admission MIMIC-IV cohort show that SW achieves higher concordance than the evaluated baseline methods. Ablation studies examine the contributions of retrieval and trajectory reasoning, while scenario analyses link alternative future trajectories to differences in predicted survival. In addition to predicting survival, SW generates interpretable and traceable patient trajectories that support exploration of alternative disease-progression scenarios and hypothesis generation about patients' future conditions.

1 INTRODUCTION

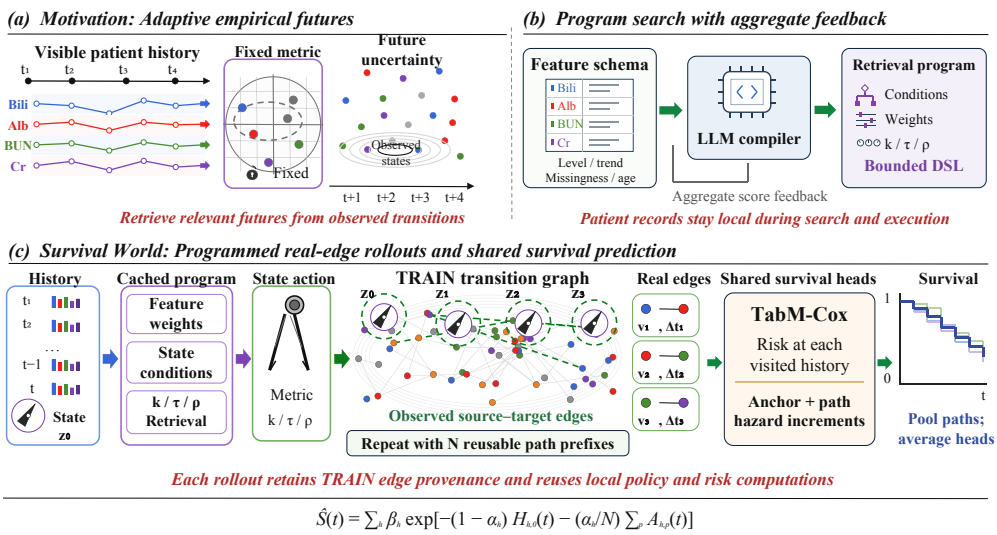
A longitudinal patient record contains irregular observations of a changing process (Che et al., 2018). Survival prediction models summarize this history to estimate individual risk or survival probabilities (Haider et al., 2020). However, these outputs alone do not reveal which possible future developments support a forecast or how to trace them back to observed evidence. We therefore ask a complementary question: can a predictor retrieve observed patient transitions, compose them into explicit future paths, and preserve the provenance of the resulting survival estimate?

Existing approaches commonly formulate prognosis through risk scores or survival distributions (Cox, 1972; Haider et al., 2020). Although longitudinal models can capture temporal information (Zeng et al., 2025), predicting an endpoint does not necessarily provide an explicit account of how a patient's condition may evolve before that endpoint. Therefore, exploring how patient states may evolve over time can provide supporting evidence for survival forecasts, making the resulting predictions more interpretable.

World models offer a promising approach to this challenge by simulating future states (Ha & Schmidhuber, 2018). In healthcare, Foresight generates clinical event timelines from electronic health records (Kraljevic et al., 2024), while Medical World Model simulates tumor evolution under different treatment conditions (Yang et al., 2025). These approaches demonstrate the potential of modeling disease progression. However, learning generative dynamics in world models often requires large-scale training data and substantial computational resources. Applying these models to new disease settings may also require further adaptation and validation.

We introduce SURVIVAL WORLD (SW), a programmable empirical world model for survival prediction. A large language model translates feature descriptions and aggregate evaluation feedback into a

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.



$$\hat{S}(t) = \sum_h \beta_h \exp[-(1-\alpha_h) H_{h0}(t) - (\alpha_h/N) \sum_p A_{hp}(t)]$$

Figure 1: **Survival World.** (a) Observed patient histories provide evidence for alternative futures. (b) Feature descriptions and aggregate feedback help a language model design retrieval rules. (c) These rules select observed transitions from training patients to build future trajectories. Shared survival predictors estimate risk along these trajectories and combine their predictions into a survival curve. Each step remains linked to its supporting training observations.

bounded retrieval program. Starting from a patient's observed history, the program retrieves relevant transitions from a graph of training histories and repeatedly advances to their observed successor states, constructing multiple alternative future trajectories. Shared survival predictors evaluate the histories visited along each trajectory, and their hazard contributions are aggregated across trajectories to produce a patient-specific survival curve.

Our contributions are threefold. First, we introduce training-free empirical trajectory construction that composes observed patient transitions into alternative futures, enabling trajectory exploration without fitting a new generative dynamics network. Second, we develop an inspectable retrieval program and retain the supporting observations for every transition, allowing survival forecasts to be traced to explicit patient histories. Third, we connect trajectory-based survival prediction with comparative scenario analysis, enabling users to examine the patient-state patterns associated with different forecasts and generate hypotheses about disease progression. Experiments on PBCseq and a second-admission MIMIC-IV cohort show higher concordance than the evaluated baselines, while ablation studies clarify the contributions of retrieval, trajectory reasoning, and training supervision.

2 Related Work

Survival prediction and longitudinal modeling. Traditional survival models estimate risk from patient features. Cox regression assumes proportional hazards (Cox, 1972), while random survival forests capture nonlinear relationships (Ishwaran et al., 2008). More flexible methods use multitask learning and neural networks to predict risk or survival probabilities, including MTLR, DeepSurv, DeepHit, discrete-time hazard models, and SurvTRACE (Yu et al., 2011; Katzman et al., 2018; Lee et al., 2018; Gensheimer & Narasimhan, 2019; Wang & Sun, 2022). Longitudinal methods additionally model how patient information changes over time. Neural controlled differential equations handle irregular observations (Kidger et al., 2020), and TrajSurv learns continuous patient representations to explain progression patterns associated with survival (Zeng et al., 2025). SW builds on this interest in temporal evolution by assembling observed patient transitions into future paths whose supporting evidence can be inspected.

World models, retrieval, and program search. World models learn how states change and use this knowledge to simulate possible futures (Ha & Schmidhuber, 2018). Medical examples include

Foresight for clinical event timelines and Medical World Model for tumor evolution (Kraljevic et al., 2024; Yang et al., 2025). Retrieval-based methods instead draw on stored evidence: retrieval-augmented generation retrieves passages to support language generation (Lewis et al., 2020), while SW retrieves patient transitions to construct trajectories. Program-search methods use language models to propose solutions and improve them through evaluation feedback, as in OPRO and FunSearch (Yang et al., 2024; Romera-Paredes et al., 2024). SW uses this approach to design rules for selecting relevant transitions. Unlike counterfactual survival explanations that modify patient inputs (Kovalev et al., 2021), SW keeps the observed history fixed and compares alternative future paths.

3 Empirical Survival World

3.1 Prediction setting and observable state

For patient i, let $\mathcal{H}_i = \{(x_{ij}, a_{ij}, o_{ij})\}_{j=1}^{m_i}$ denote the history visible at the prediction landmark. Here $x_{ij} \in \mathbb{R}^d$ is a measurement vector, a_{ij} is its observation time, and o_{ij} records missingness and available measurement provenance. We predict residual event time using an observed time y_i and censoring indicator δ_i. Times retain their physical units; histories are truncated at their stated landmarks.

A deterministic summary $\phi(\mathcal{H}_i)$ combines current values, historical means, recent slopes, current and historical missingness, measurement and availability ages, reuse indicators, corresponding known-status flags, and three timing descriptors. Its dimension is $11d + 3$: 157 for PBCseq and 135 for MIMIC-IV. Imputation uses training-visible medians. Unknown provenance has an explicit known-status channel rather than an inferred freshness value. For retrieval, standardized levels ℓ, velocities v, and summary features h form

$$g(\mathcal{H}) = \left[\ell/\sqrt{d};\ v/\sqrt{d};\ h/\sqrt{11d+3}\right]. \tag{1}$$

3.2 An empirical transition graph

Each node stores an actual training prefix. For consecutive observations of a training patient, we create an edge $e = (u_e, v_e, \Delta_e)$ with $\Delta_e > 0$. The graph contains only training patients; its node histories precede their scalar endpoints. Retrieval uses state and observation features, without endpoint labels in its distance. An edge therefore provides both a candidate successor and the elapsed time over which that transition occurred.

At rollout state z, an exact geometric nearest-neighbor index returns up to 800 source edges. We rerank this shortlist with the program-defined distance, exclude the original query identity and the current donor, and sample an eligible edge. The rollout then adopts the target node's actual prefix. The query identity remains fixed even as donor histories change. This creates a compositional empirical dynamics model: an observed edge supports each jump, while the assembled multi-donor path represents a model-generated scenario.

3.3 Adaptive Retrieval Program

The policy π maps the current state to an action containing feature weights w, interaction groups, neighborhood size k, temperature τ, a local semantic coefficient b, and an optional geometric fallback. Conditions can inspect standardized level bins, trend bins, missingness, measurement age, availability age, and reuse status. Branches use conjunction or disjunction; all matching branches execute in order. Multiplicative weights are clipped to $[0.05, 8]$, and the last matching scalar assignment takes effect. Bounds on program size and numerical parameters make candidate validation explicit (Appendix C).

Let m_{ej} indicate that feature j is observed in both the query state and candidate source and is enabled by the action. Let q_{ej} indicate that both trends are known. The semantic distance follows the implemented masked normalization:

$$D_{\mathrm{sem}}(z, u_e) = \frac{\sum_j w_j m_{ej}\{(\ell_{zj} - \ell_{ej})^2 + q_{ej}(v_{zj} - v_{ej})^2\}}{\max(\sum_j w_j m_{ej}, 1)} + I_\pi(z, u_e), \tag{2}$$

where I_π adds 0.25 times the squared mean level difference for each fully observed interaction group. With external search coefficient η and local coefficient $b_\pi(z)$, we use $\rho(z) = \eta b_\pi(z)$ and

$$D_\pi = (1-\rho)\|g(z) - g(u_e)\|_2^2 + \rho D_{\text{sem}}. \tag{3}$$

Candidates without semantic support, or actions requesting fallback, retain the geometric distance. On the eligible top-k set $\mathcal{E}_k(z)$, the transition kernel is

$$p_\pi(e \mid z) = \frac{\exp[-\sqrt{D_\pi(z,u_e)}/(\tau s_z)]}{\sum_{e' \in \mathcal{E}_k(z)} \exp[-\sqrt{D_\pi(z,u_{e'})}/(\tau s_z)]}, \quad s_z = \text{median}_{e \in \mathcal{E}_k(z)} \sqrt{D_\pi(z,u_e)} + 10^{-12}. \tag{4}$$

Semantic reranking is confined to the geometric shortlist. State dependence comes from local program execution, so no network request is needed when a patient moves to a new rollout state.The program remains fixed during inference, but its rules adapt retrieval to each visited state; Appendix C provides concrete policy examples and specifies the predicates and bounds.

Fitted retrieval geometry. The selected configurations add a prognostic coordinate derived from TRAIN-fitted shared heads. The negative integrated head survival is standardized across training nodes and clipped to $[-10, 10]$. Appending its value scaled by $\sqrt{\omega}$ to g changes reranking, while the original 800-edge shortlist is retained. The PBCseq representative uses $\omega = 4$; the MIMIC-IV representative uses $\omega = 0$. Local numerical tuning also scales k and temperature and transforms positive feature weights by a fixed power. The prognostic-coordinate weight, head fits, pooling, and sampling scheme are shared with the matched geometric control. No query endpoint enters retrieval at inference.

3.4 Shared survival heads and path pooling

We fit TabM-Cox heads on training summaries, use calibration data for early stopping, and estimate cumulative baseline hazards with tied-time Breslow handling (Breslow, 1974). Let $H_{h,z}(t)$ denote the cumulative hazard from head h at history z. For path p with states $z_{p,r}$ and elapsed transition times $s_{p,r}$, define

$$A_{h,p}(t) = \sum_{r: s_{p,r} < t} \left[H_{h,z_{p,r}}(\min(t, s_{p,r+1})) - H_{h,z_{p,r}}(s_{p,r}) \right]_+ . \tag{5}$$

The time argument is global elapsed prediction time. Execution stops after at most 60 transitions and extends any remaining horizon with the current node's head. Let $S_{h,0}(t) = \exp[-H_{h,z_0}(t)]$ be the query anchor. We pool paths using

$$S_{h,q}(t) = S_{h,0}(t)^{1-\alpha_h} \begin{cases} \left[\frac{1}{N} \sum_{p=1}^N \exp(-q\alpha_h A_{h,p}(t)) \right]^{1/q}, & q > 0, \\ \exp\left[-\frac{\alpha_h}{N} \sum_{p=1}^N A_{h,p}(t) \right], & q = 0. \end{cases} \tag{6}$$

The $q = 1$ case averages path survival probabilities after anchor mixing; the continuous $q = 0$ limit averages hazards. The final prediction averages head curves:

$$\widehat{S}(t \mid \mathcal{H}) = \sum_{h=1}^J \beta_h S_{h,q}(t), \qquad \beta_h \geq 0, \quad \sum_h \beta_h = 1. \tag{7}$$

The representative configurations use two equally weighted heads, 128 paths, and $q = 0$; Table 6 specifies the cohort-specific anchor mixtures. The scenario analysis uses a fixed probability-pooling configuration with 160 paths (Appendix E). Ranking uses the negative survival integral over the evaluation grid.

3.5 Program search, execution, and calibration

Program search. The language model (Appendix F.1) proposes retrieval programs from feature descriptions, the retrieval procedure, and aggregate evaluation feedback. Each proposal must follow a restricted language that defines the conditions and actions available to the program. This allows the language model to adjust retrieval rules without generating patient measurements or accessing patient outcomes during inference. Search evaluates these programs together with survival-head combinations, aggregation choices, and retrieval parameters. Candidate evaluations reuse fitted survival heads, avoiding the need to retrain predictors for each proposal.

Table 1: Cohort and graph sizes after endpoint exclusions. Patient counts are followed by event counts in parentheses. Graph nodes and edges belong exclusively to training patients.

Dataset	TRAIN	CAL	TEST	Graph nodes	Graph edges
PBCseq	199 (90)	50 (22)	63 (28)	1,244	1,045
MIMIC-IV	7,968 (3,134)	2,658 (1,022)	2,713 (1,072)	31,184	23,216

Efficient execution. Retrieval candidates, program outputs, and survival-head predictions are cached to avoid repeated computation, while the supporting transitions are retained so that trajectories can be replayed and inspected. The representative configurations use 128 paths with scrambled Sobol sampling (Joe & Kuo, 2008; SciPy Community, 2024). We repeat simulations with different seeds to assess sampling variability and use the same seeds for paired comparisons.

Probability calibration. To improve agreement between predicted survival probabilities and observed outcomes, we select a positive exponent γ that minimizes the integrated Brier score on the calibration set (CAL). We then transform the predicted survival curve as $\widehat{S}_{\mathrm{cal}}(t) = \widehat{S}(t)^{\gamma}$, using the same fitted exponent for all test runs. We report results before and after calibration separately, because this transformation can change both probability accuracy and rankings based on the area under the survival curve. CAL is also used for early stopping when training the survival heads.

3.6 CONSTRUCTION PROPERTIES AND INSPECTABLE SCENARIOS

Proposition 1 (Empirical support and valid survival) *Suppose every selected edge has positive duration and every shared head has a nonnegative, nondecreasing cumulative hazard with $H_{h,z}(0) = 0$. Every simulated successor is an observed TRAIN prefix with an identifiable supporting edge. For $q \geq 0$, $\alpha_h \in [0, 1]$, and nonnegative normalized head weights, Equation 7 starts at one, remains in $[0, 1]$, and is nonincreasing. Positive-exponent calibration preserves these properties.*

Appendix A proves these construction properties. For a fixed query, ranking paths by their predicted survival area permits comparison of upper- and lower-quartile continuations and their retrieved node features. This connects alternative survival curves to the observed transitions composing each predictive scenario (Appendix E).

4 EXPERIMENTAL PROTOCOL

4.1 DATASETS AND PREDICTION LANDMARKS

Table 1 summarizes the completed cohorts. PBCseq contains serial observations from the primary biliary cirrhosis data distributed with the survival package (Murtaugh et al., 1994; Therneau, 2026). We use 14 clinical features and the last retained pre-endpoint observation as each patient's landmark. This is a retrospective last-record prediction task. Per-assay measurement age, availability age, and reuse are unknown and are represented accordingly.

MIMIC-IV (Johnson et al., 2023) uses an exact second-admission landmark and 12 features. Both the earlier-admission node and the current node use observations available as of their own timestamps. Primary-assay identifiers and units are preserved, and both chart and storage availability are checked. All eligible future transitions of training patients can contribute to the training graph; evaluation patients expose only their two landmark-visible nodes. Eight unresolved endpoint records are excluded across the three partitions. Appendix B details endpoint construction and the inherited cohort definition.

4.2 COMPARATORS AND METRICS

We compare ten baseline methods: CoxPH (Cox, 1972), gradient boosting survival (GBS) (Hothorn et al., 2006), random survival forest (RSF) (Ishwaran et al., 2008), GRU-LogisticHazard (Gensheimer & Narasimhan, 2019), SurvTRACE (Wang & Sun, 2022), TabM-Cox (Gorishniy et al., 2025; Cox, 1972), a local NCDE state-alignment baseline (Kidger et al., 2020), DeepSurv (Katzman et al., 2018),

single-risk DeepHit (Lee et al., 2018), and neural MTLR (Fotso, 2018). DeepSurv, DeepHit, and neural MTLR use pycox implementations (Kvamme et al., 2019); implementation details are given in Appendix F. The NCDE baseline is not an exact reproduction of TrajSurv (Zeng et al., 2025).

We tune the hyperparameters of each baseline separately for each cohort. Diagnostic comparisons use matched anchors and geometric rollouts with the same fitted heads and numerical predictor settings as SW. Geometry removes the complete retrieval program; an additional PBCseq control disables semantic distance while preserving program actions. These comparisons distinguish the contribution of the complete retrieval policy from that of semantic distance alone.

Evaluation. TRAIN fits preprocessing and model parameters; CAL supports early stopping and probability calibration. Scores from the partition named TEST were used for evaluation. Search budgets differ: program search evaluates substantially more candidates than baseline tuning, often reusing fitted heads and paths. We retain the original patient splits and search seed and evaluate fixed SW configurations with five fresh simulation seeds on the same patients.

Metrics. Harrell-style C-index (Harrell et al., 1996) is the primary evaluation metric. IPCW Brier score and integrated Brier score (IBS) (Graf et al., 1999; Gerds & Schumacher, 2006) assess survival probability accuracy using the censoring distribution estimated from TRAIN. C-index is computed with scikit-survival (Pölsterl, 2020); Brier score and IBS use a common IPCW implementation. All methods are evaluated on a grid of 50 time points from 120 days to $\max(180, Q_{0.9}^{\text{train}})$, where $Q_{0.9}^{\text{train}}$ is the 90th percentile of training residual survival times. Brier score is evaluated at the median training residual survival time, clipped to the grid bounds. For C-index, the risk score is the negative area under the predicted survival curve over this grid. Each reported row contains all metrics from the same configuration and calibration state.

Baseline and ablation comparisons. Table 2 compares the baseline methods with SW and SW + CAL. For each predictor, we report all metrics from the configuration selected by C-index. Additional tables examine retrieval controls, individual PBCseq components, and training supervision. Retrieval and calibration comparisons use fixed SW configurations with 128 paths per head and the same simulation seeds across paired conditions. Appendix F provides implementation details and selected configurations.

5 RESULTS

5.1 COMPARISON WITH BASELINE METHODS

Table 2 compares SW and SW + CAL with ten baseline methods. SW achieves the highest reported C-index on both cohorts. On PBCseq, its C-index exceeds the strongest baseline by 0.00750, and its Brier score is 8.2% lower than the best baseline for that metric. On MIMIC-IV, the C-index improvement over the strongest baseline is small. No method performs best on all probability metrics: neural MTLR achieves the lowest PBCseq IBS, while SurvTRACE and TabM-Cox achieve the lowest MIMIC-IV Brier score and IBS, respectively. SW + CAL applies probability calibration to the same fitted heads and trajectories.

5.2 COMPONENT CONTRIBUTIONS AND SIMULATION VARIABILITY

The upper block of Table 3 isolates retrieval effects on the final predictors. PBCseq C-index rises from 0.8782 for the anchor to 0.8910 with geometric rollouts and 0.8925 with SW. On MIMIC-IV the corresponding values are 0.7332, 0.7331, and 0.7336. SW exceeds geometry in four of five PBCseq seeds and all five MIMIC-IV seeds (Figure 4), showing consistent ranking across the paired simulation draws.

Geometry removes the entire retrieval program, including adaptive neighborhood size and temperature. A narrower PBCseq control disables semantic distance while retaining program actions. SW-minus-control differences are -0.000511 in mean C-index and -0.000794 in IBS; C-index improves in two seeds, declines in two, and ties in one. Thus, the gain over Geometry does not establish a benefit from semantic distance alone.

Table 2: **Comparison with baseline methods.** Ten external baselines are compared with the final SW predictor and its CAL-fitted version on the same TRAIN/CAL/TEST patients, visible histories, grid, and metrics. Baselines report selected single runs; SW uses two heads and 128 paths, averaging five simulation-seed scores. Bold marks the best displayed value across all methods, including ties at displayed precision. The rule separates the proposed method from external baselines. The shared evaluation protocol is given in Section 4.2.

	PBCseq			MIMIC-IV		
Predictor / condition	C-index ↑	Brier ↓	IBS ↓	C-index ↑	Brier ↓	IBS ↓
SurvTRACE (official)	0.885009	0.476201	0.509213	0.733221	**0.153839**	0.183467
DeepHit (single risk)	0.885009	0.212240	0.196490	0.725027	0.169192	0.206254
Random survival forest	0.880750	0.107904	0.113231	0.732133	0.156996	0.186051
TabM–Cox	0.879046	0.101556	0.114000	0.733404	0.154139	**0.182441**
CoxPH	0.873935	0.228367	0.256750	0.721218	0.159675	0.190761
GRU–LogisticHazard	0.870528	0.114022	0.125666	0.726679	0.160782	0.192730
NCDE state alignment	0.867973	0.134619	0.175729	0.716828	0.158377	0.189761
DeepSurv (pycox)	0.867121	0.104116	0.120780	0.727876	0.156812	0.185397
Neural MTLR	0.867121	0.106891	**0.091304**	0.727661	0.155534	0.186056
Gradient boosting survival	0.865417	0.090758	0.113620	0.729244	0.155832	0.185804
SW (ours)	**0.892504**	**0.083303**	0.127555	**0.733638**	0.154558	0.184575
SW + CAL (ours)	0.891993	0.096022	0.097329	**0.733638**	0.154599	0.183902

Table 3: **Retrieval and component controls.** Anchor only predicts from the query history with the same fitted heads, without rollouts. Geometry retains rollouts but removes the retrieval program. Upper block: final two-head, 128-path models; rollout scores average five seeds. Lower block: original single-head SW and eight variants (PBCseq $N = 40$, $\alpha = 1$; MIMIC-IV $N = 10$, $\alpha = 0.5$), evaluated in single runs. Anchors are deterministic. Compare within each block, not across blocks. Appendix C.1 defines every variant and its fixed settings, including the fixed template's distinct retrieval defaults.

	PBCseq			MIMIC-IV		
Predictor / condition	C-index ↑	Brier ↓	IBS ↓	C-index ↑	Brier ↓	IBS ↓
Final predictor: matched retrieval controls						
Final SW	0.892504	0.083303	0.127555	0.733638	0.154558	0.184575
Final anchor only	0.878194	0.100968	0.113643	0.733241	0.154035	0.182247
Final geometric rollouts	0.890971	0.083988	0.125339	0.733145	0.154791	0.184941
Original single-head model: component controls						
Original frozen SW	0.887564	0.195315	0.227169	0.734393	0.155099	0.185929
Anchor only	0.868825	0.196319	0.224462	0.732795	0.154373	0.182840
One path	0.851789	0.201236	0.227584	0.710609	0.160486	0.193721
Mean hazard aggregation	0.887564	0.195175	0.226924	0.734149	0.155173	0.186743
Geometric retrieval	0.878194	0.195433	0.227892	0.732324	0.155225	0.187303
Fixed template	0.878194	0.195162	0.228464	0.731447	0.155387	0.186157
Permuted program	0.880750	0.195532	0.227322	0.734151	0.155093	0.185964
No dynamic branches	0.885009	0.195319	0.227158	0.734233	0.155093	0.185929
No feature reweighting	0.879046	0.195013	0.227649	0.734404	0.155100	0.185887

The lower block of Table 3 compares the original single-head SW with eight variants defined in Appendix C.1. Disabling trajectories or using only one path reduces C-index on both cohorts. PBCseq also benefits from dynamic branches and reweighting in this frozen setup; MIMIC-IV reweighting has no positive ranking effect. These comparisons use the original frozen configuration, separately from the final predictor above.

Table 4 adds targeted controls on the final PBCseq predictor. The mean across-patient variance of predicted survival area over five simulation seeds is 383.54 days^2 with scrambled Sobol draws and 1113.05 days^2 with IID draws, a 65.5% reduction. The Sobol-to-IID variance ratio is 0.345, with a conditional paired-patient bootstrap interval of $[0.188, 0.623]$. This variance reduction measures simulation stability at fixed patients and settings. Mean C-index is 0.892504 with Sobol and 0.888416 with IID. Removing the prognostic coordinate gives 0.884327 under the same fixed predictor.

Table 4: **Additional controls on the final PBCseq predictor.** All rows retain its two heads, $N = 128$, and $q = 0$, averaging five fresh simulation seeds. The last row removes semantic distance but preserves program actions.

Condition	C-index ↑	Brier ↓	IBS ↓
SW	0.892504	0.083303	0.127555
SW (IID)	0.888416	0.083726	0.130185
SW (No risk coordinates)	0.884327	0.088551	0.120335
No semantic distance	0.893015	0.083128	0.128349

Table 5: **Training-supervision controls.** Each TRAIN patient contributes two rows: two copies of the same query, or one query plus one observed prefix. Two head configurations are refitted per condition, with unchanged CAL/TEST queries. Anchors are deterministic; SW and geometric rollouts average five seeds. This experiment has its own frozen retrieval setup (Appendix D); it measures training effects separately from the primary method comparison.

	PBCseq			MIMIC-IV		
Predictor / condition	C-index ↑	Brier ↓	IBS ↓	C-index ↑	Brier ↓	IBS ↓
Duplicated-query training						
Anchor	0.881601	0.090238	0.105521	0.731770	0.154447	0.182731
Geometry	0.882964	0.077827	0.133761	0.732102	0.154572	0.183472
SW	0.884157	0.078159	0.133555	0.732056	0.154553	0.183435
Query-plus-prefix training						
Anchor	0.877342	0.101119	0.112026	0.737609	0.153719	0.180879
Geometry	0.886542	0.085534	0.118138	0.737432	0.154360	0.183598
SW	0.885009	0.085298	0.119976	0.737420	0.154290	0.183371

5.3 Calibration improves integrated probability quality

Ranking and probability quality capture different aspects of survival prediction (Haider et al., 2020; Van Calster et al., 2019). CAL-fitted scaling reduces PBCseq IBS from 0.1276 to 0.0973, a 23.7% decrease, across all five seeds. Its C-index changes from 0.8925 to 0.8920 and its single-horizon Brier score increases. MIMIC-IV IBS decreases from 0.1846 to 0.1839, again in every seed, but remains worse than the matched anchor's 0.1822. The method therefore supports a useful calibration step, while ranking, integrated error, and a single-horizon error expose distinct tradeoffs. Appendix D gives the calibration grid and paired changes (Figure 5).

5.4 Separating training gains from trajectory gains

Table 5 matches two training rows per patient, head configurations, queries, and evaluation settings. On MIMIC-IV, prefix supervision improves anchor C-index by 0.005839, with a conditional paired-patient bootstrap interval of $[0.002853, 0.009025]$. The prefix-trained anchor exceeds its SW counterpart on all three metrics, and CAL-selected fusion assigns zero trajectory weight. These results attribute the gain primarily to training. PBCseq shows no consistent prefix benefit. The duplicated-query condition controls for the number of training rows.

5.5 Tracing alternative empirical futures

The empirical graph makes predictions inspectable beyond their aggregate scores. Figure 2 shows alternative survival curves from the same observed history. Each simulated transition retains a real TRAIN edge; across the frozen scenario analysis, PBCseq paths visit 989 edges from 181 donors and MIMIC-IV paths visit 20,768 edges from 6,822 donors.

Figure 3 links alternative predictions to their supporting observations. Higher-area PBCseq paths retrieve first targets with lower bilirubin and higher albumin; MIMIC-IV differences include lower RDW and BUN and higher hemoglobin and RBC. Displaying all features reveals both pronounced

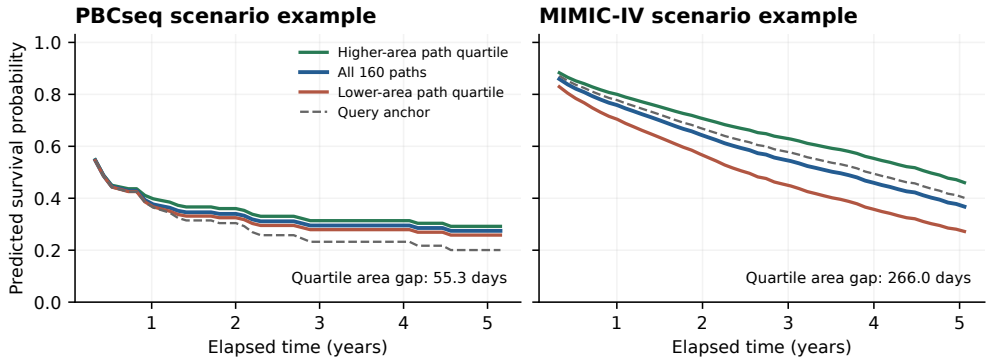


Figure 2: **Inspectable alternative futures.** A fixed probability-pooling configuration generates 160 paths per query, separate from the mean-hazard predictor in Table 2. Upper- and lower-area path quartiles describe alternative model-generated continuations of the same observed history. Examples were selected near the cohort median path-area spread without using outcomes. Curves are not confidence intervals or treatment effects.

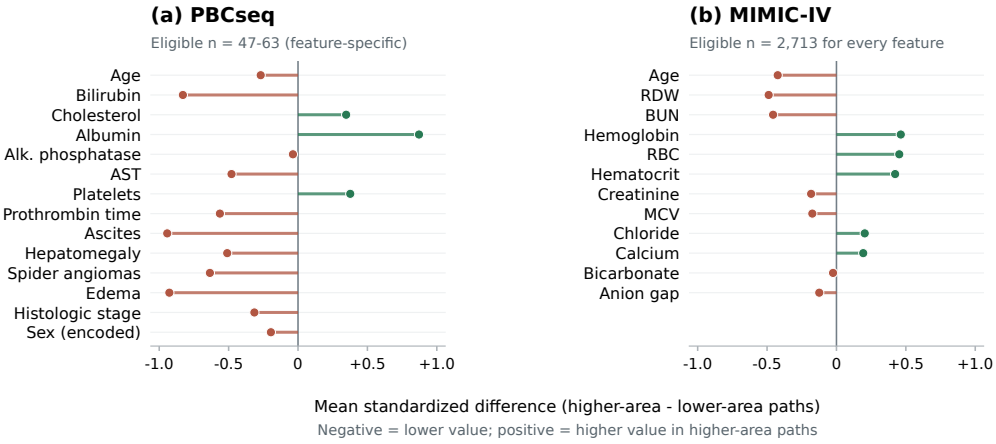


Figure 3: **Observed profiles behind alternative futures.** Mean within-query standardized differences between upper- and lower-area quartiles' first targets, from the same 160-path analysis as Figure 2. All 26 features are shown (14 PBCseq, 12 MIMIC-IV); eligible query counts vary by feature. These retrieved TRAIN-node associations describe predictive scenarios, not intervention effects.

and small contrasts. Grouping by predicted area makes these descriptive scenario associations; Appendix E details eligibility and aggregation.

6 Conclusion

We introduced Survival World (SW), a programmable empirical world model that constructs alternative patient trajectories from observed transitions without training a new dynamics network. Shared survival predictors combine these trajectories into forecasts, while retained transition evidence makes predictions traceable. Experiments on PBCseq and MIMIC-IV show higher concordance than the evaluated baselines, with ablations examining retrieval, trajectory, and supervision effects. Beyond survival prediction, SW supports inspection of alternative disease-progression scenarios and hypothesis generation. Future work will assess generalization to new cohorts.

AI USE STATEMENT

In this work, we used generative AI tools for finding relevant baseline methods, assisting with the implementation of parts of the code, and checking mathematical formulas. As described in the main text, a large language model was also used to propose retrieval programs from feature descriptions and aggregate evaluation feedback. Additionally, we used generative AI tools to polish the manuscript for clarity and readability. The authors reviewed the AI-assisted text, checked the identified baseline methods against their original sources, and verified the assisted code and mathematical formulas. The authors take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This study uses the PBCseq and MIMIC-IV datasets for retrospective survival prediction research. Patient-level records remain local, and restricted MIMIC-IV data and patient-level derived artifacts are excluded from the public supplement.

REPRODUCIBILITY STATEMENT

The accompanying materials provide implementation and plotting code, selected configurations, and aggregate results. Appendix F provides implementation details for the neural baselines.

REFERENCES

N. Breslow. Covariance analysis of censored survival data. *Biometrics*, 30(1):89–99, 1974.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8:6085, 2018. doi: 10.1038/s41598-018-24271-9.

D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B*, 34 (2):187–202, 1972. doi: 10.1111/j.2517-6161.1972.tb00899.x.

Stephane Fotso. Deep neural networks for survival analysis based on a multi-task framework. *arXiv preprint arXiv:1801.05512*, 2018.

Michael F. Gensheimer and Balasubramanian Narasimhan. A scalable discrete-time survival model for neural networks. *PeerJ*, 7:e6257, 2019. doi: 10.7717/peerj.6257.

Thomas A. Gerds and Martin Schumacher. Consistent estimation of the expected Brier score in general survival models with right-censored event times. *Biometrical Journal*, 48(6):1029–1040, 2006. doi: 10.1002/bimj.200610301.

Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. TabM: Advancing tabular deep learning with parameter-efficient ensembling. In *International Conference on Learning Representations*, 2025.

Erika Graf, Claudia Schmoor, Willi Sauerbrei, and Martin Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in Medicine*, 18(17–18):2529–2545, 1999.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Humza Haider, Bret Hoehn, Sarah Davis, and Russell Greiner. Effective ways to build and evaluate individual survival distributions. *Journal of Machine Learning Research*, 21(85):1–63, 2020.

Frank E. Harrell, Jr., Kerry L. Lee, and Daniel B. Mark. Multivariable prognostic models: Issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in Medicine*, 15(4):361–387, 1996.

Torsten Hothorn, Peter Bühlmann, Sandrine Dudoit, Annette Molinaro, and Mark J. van der Laan. Survival ensembles. *Biostatistics*, 7(3):355–373, 2006. doi: 10.1093/biostatistics/kxj011.

Hemant Ishwaran, Udaya B. Kogalur, Eugene H. Blackstone, and Michael S. Lauer. Random survival forests. *The Annals of Applied Statistics*, 2(3):841–860, 2008. doi: 10.1214/08-AOAS169.

Stephen Joe and Frances Y. Kuo. Constructing Sobol' sequences with better two-dimensional projections. *SIAM Journal on Scientific Computing*, 30(5):2635–2654, 2008. doi: 10.1137/070709359.

Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li-wei H. Lehman, Leo A. Celi, and Roger G. Mark. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific Data*, 10:1, 2023. doi: 10.1038/s41597-022-01899-x.

Jared L. Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. DeepSurv: Personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18:24, 2018. doi: 10.1186/s12874-018-0482-1.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Maxim Kovalev, Lev Utkin, Frank Coolen, and Andrei Konstantinov. Counterfactual explanation of machine learning survival models. *Informatica*, 32(4):817–847, 2021. doi: 10.15388/21-INFOR468.

Zeljko Kraljevic, Dan Bean, Anthony Shek, Rebecca Bendayan, Harry Hemingway, Joshua Au Yeung, Alexander Deng, Alfred Baston, Jack Ross, Esther Idowu, James T. Teo, and Richard J. B. Dobson. Foresight: A generative pretrained transformer for modelling of patient timelines using electronic health records: A retrospective modelling study. *The Lancet Digital Health*, 6(4):e281–e290, 2024. doi: 10.1016/S2589-7500(24)00025-6.

Håvard Kvamme, Ørnulf Borgan, and Ida Scheel. Time-to-event prediction with neural networks and Cox regression. *Journal of Machine Learning Research*, 20(129):1–30, 2019.

Changhee Lee, William R. Zame, Jinsung Yoon, and Mihaela van der Schaar. DeepHit: A deep learning approach to survival analysis with competing risks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 2314–2321, 2018. doi: 10.1609/aaai.v32i1.11842.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

P. A. Murtaugh, E. R. Dickson, G. M. Van Dam, M. Malinchoc, P. M. Grambsch, A. L. Langworthy, and C. H. Gips. Primary biliary cirrhosis: Prediction of short-term survival based on repeated patient visits. *Hepatology*, 20(1 Pt 1):126–134, 1994. doi: 10.1016/0270-9139(94)90144-9.

Sebastian Pölsterl. scikit-survival: A library for time-to-event analysis built on top of scikit-learn. *Journal of Machine Learning Research*, 21(212):1–6, 2020.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2024. doi: 10.1038/s41586-023-06924-6.

SciPy Community. scipy.stats.qmc.Sobol. SciPy 1.14.1 Reference Manual, 2024. URL `https://docs.scipy.org/doc/scipy-1.14.1/reference/generated/scipy.stats.qmc.Sobol.html`. Accessed 2026-09-25.

Terry M. Therneau. Mayo clinic primary biliary cirrhosis, sequential data. R survival package documentation, 2026. URL `https://stat.ethz.ch/R-manual/R-devel/library/survival/html/pbcseq.html`. Accessed 2026-09-25.

Ben Van Calster, David J. McLernon, Maarten van Smeden, Laure Wynants, and Ewout W. Steyerberg. Calibration: The Achilles heel of predictive analytics. *BMC Medicine*, 17:230, 2019. doi: 10.1186/s12916-019-1466-7.

Zifeng Wang and Jimeng Sun. SurvTRACE: Transformers for survival analysis with competing events. In *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. Association for Computing Machinery, 2022. doi: 10.1145/3535508.3545521. Article 49.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, 2024.

Yijun Yang, Zhao-Yang Wang, Qiuping Liu, Shuwen Sun, Kang Wang, Rama Chellappa, Zongwei Zhou, Alan Yuille, Lei Zhu, Yu-Dong Zhang, and Jieneng Chen. Medical world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8319–8329, 2025.

Chun-Nam Yu, Russell Greiner, Hsiu-Chin Lin, and Vickie Baracos. Learning patient-specific cancer survival distributions as a sequence of dependent regressors. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

Sihang Zeng, Lucas Jing Liu, Jun Wen, Meliha Yetisgen, Ruth Etzioni, and Gang Luo. TrajSurv: Learning continuous latent trajectories from electronic health records for trustworthy survival prediction. In *Proceedings of the 10th Machine Learning for Healthcare Conference*, volume 298 of *Proceedings of Machine Learning Research*, 2025.

A Construction Properties

Proof. Each jump selects a TRAIN edge and adopts its actual target prefix. Induction establishes step-level provenance; positive durations and the 60-transition cap ensure finite execution. A complete path may combine donors.

Every term in $A_{h,p}(t)$ is initially zero and nondecreasing, so $A_{h,p}(0) = 0$. For $q > 0$, each $\exp[-q\alpha_h A_{h,p}(t)]$ starts at one, stays in $[0, 1]$, and is nonincreasing. Its path average and positive $1/q$ power retain these properties. Multiplication by the nonnegative, nonincreasing anchor factor yields the same result for $S_{h,q}$. For $q = 0$, the average path hazard is nondecreasing, and exponentiating its negative gives the same conclusion. Convex head mixtures and positive-exponent calibration preserve all three survival-curve properties. □

These are shape and provenance guarantees, not guarantees of predictive accuracy or causal validity. The representative configurations use scrambled Sobol draws, whose paths are not independent Monte Carlo samples. We therefore report measured variation across scrambles rather than applying an independent-path concentration bound. Grid interpolation preserves nondecreasing hazards, with zero cumulative hazard to the left of the first grid point.

B Cohort Construction and Observation Context

PBCseq. The cohort contains 312 patients, 140 events, and 1,945 visible nodes. Its feature list is age, bilirubin, cholesterol, albumin, alkaline phosphatase, AST, platelets, prothrombin time, ascites, hepatomegaly, spiders, edema, stage, and sex. The inherited patient split has 199 training, 50 calibration, and 63 evaluation patients. Histories preserve observed time intervals. The query is the last retained pre-endpoint record, making this a retrospective last-record prediction task. Laboratory provenance unavailable in this source is explicitly unknown, including measurement age, storage availability age, and reuse status. A branch requiring known reuse cannot activate on these unknown values.

MIMIC-IV. After exclusions, the cohort contains 13,339 patients and 5,228 events. There are 26,678 query-visible nodes: two per patient. The training graph additionally includes valid later training observations, for 31,184 nodes and 23,216 positive-duration edges. The 12 features are age, RDW, BUN, hemoglobin, RBC, hematocrit, creatinine, MCV, chloride, calcium, bicarbonate, and anion gap. Each laboratory follows the original primary item identifier and explicit unit mapping. Missing units are not guessed and alternate assays are not merged. Availability requires the relevant chart and storage timestamps to meet the node's exact admission cutoff. Age is based on the documented deidentified anchor-age construction.

Cohort membership follows the inherited prior-completed-stay diagnosis proxy; an exact historical coding timestamp is unavailable. The feature profile also has a legacy data-driven selection history. Partition membership is inherited without promoting unassigned patients, which remain quarantined. Feature extraction is separated from outcome construction. The six unresolved training endpoints, one calibration endpoint, and one evaluation endpoint are excluded from scalar survival fitting and scoring, with reasons preserved.

Mortality and observation boundary. For every MIMIC-IV patient, administrative follow-up ends at the last valid hospital discharge plus 365 elapsed days. This study convention is applied consistently to recorded deaths and to patients without a recorded death. A unique exact death time is used when consistent with the recorded death date or when that date is absent. Date-only death is represented by its calendar-day interval; eligible scalar outcomes use the interval's lower bound without adding an artificial epsilon. Intervals overlapping the landmark or administrative boundary, inconsistent death records, nonpositive residual times, and missing valid boundaries remain unresolved. Training nodes must precede the scalar endpoint. This convention does not assert complete registry capture or independent censoring; its sensitivity remains to be evaluated.

State summary and preprocessing. For d features, the summary consists of five d-vectors (current values, means, slopes, current missingness, historical missingness), three provenance values and their three known indicators ($6d$), and three scalar timing descriptors. Age-like provenance values

Table 6: Fixed numerical settings of the representative predictors. Both use two heads with equal curve weights and $q = 0$ pooling. The external fraction multiplies the program's state-dependent local fraction.

Setting	PBCseq	MIMIC-IV
Paths per head	128	128
Head-specific α	$(1, 1)$	$(0.25, 0.5)$
Prognostic-coordinate weight	4	0
Neighborhood scale	0.75	0.75
Temperature scale	0.5	1
Feature-weight power	0.5	2
Interaction-group gain	0.25	0.25
External semantic fraction	0.5	1
Trajectory sampler	Scrambled Sobol	Scrambled Sobol

use $\log(1 + x)$ when known. Timing descriptors are log-transformed visit count, history span, and final interval. Summary slopes divide by at least one day; retrieval velocities use their own positive-interval denominator. The summary and retrieval velocity channels are therefore distinct implemented features. Entirely missing training columns use zero after median handling. Endpoint labels are used to fit the survival head and define eligible histories, but are absent from the retrieval distance. When a prognostic coordinate is used, it is calculated from training-fitted heads rather than from query outcomes.

C Retrieval Language Validation and Controls

Validated program interface. A proposed program contains 1–12 branches, each with 1–8 conditions, and a default action. Level predicates use standardized bins separated at $-1.5, -0.5, 0.5, 1.5$; trend thresholds are -0.25 and 0.25. Measurement and availability age distinguish fresh (at most one day), recent (up to 30 days), stale, and unknown. Missingness and reuse have explicit status predicates. Each branch matches either all or any of its conditions. Every matching branch executes in order: weight multipliers compose, while later scalar assignments overwrite earlier ones.

The program validator bounds feature weights and multipliers to $[0.05, 8]$, neighborhood size to $[5, 100]$, temperature to $[0.1, 4]$, and local semantic fraction to $[0, 1]$. After branch execution, feature weights are clipped to $[0.05, 8]$. The numerical wrapper raises these weights to its selected power and clips again; it also rescales and rounds k into $[5, 200]$ and rescales temperature into $[0.05, 8]$. Thus program bounds and executed numerical bounds are distinct. A feature group contains 2–8 named features and contributes only when all operands are observed. Missing measurements remain excluded even if their weights increase; trend operands additionally require known trends. Programs cannot create measurements, access query outcomes, modify head parameters, or execute arbitrary code.

Representative numerical settings. Table 6 collects the fixed settings applied after local program execution. Both predictors use the same two-head pooling structure, while the selected semantic fraction, feature-weight transformation, and anchor mixture differ by cohort.

A provenance-aware PBCseq policy. All 14 default feature weights equal 1.5; there are no interaction groups. Defaults are $k = 50$, temperature 0.25, and local semantic fraction 0.5, with geometric fallback when no branch matches. In execution order, the branches are: (i) recent bilirubin measurement age, multiplying its weight by 2 and assigning $(k, \tau, b) = (40, 0.5, 0.5)$; (ii) albumin known not to be reused, multiplying its weight by 2 with the same scalar action; (iii) stable AST trend, multiplying AST by 1.5 and assigning $(30, 0.3, 0.25)$; and (iv) stale platelet availability, multiplying platelets by 2 and assigning $(50, 0.5, 0.5)$. The first, second, and fourth predicates cannot activate because the corresponding provenance is unknown in PBCseq. The stable-AST branch is the active branch in the saved evaluation; its executed neighborhood is 22 after the wrapper's rounding and its effective semantic fraction is 0.125. Inactive generated branches are retained in the program record rather than silently removed.

An executable MIMIC-IV policy. The MIMIC-IV program starts with RDW weight 1.5, other weights one, interaction group (RDW, MCV), and $(k, \tau, b) = (50, 0.5, 0.5)$. Fresh RDW together with recent MCV multiplies their weights by 2 and 1.5. Fresh chloride availability multiplies its weight by 1.5 and sets $(60, 0.25, 0.25)$; recent calcium availability multiplies its weight by 1.5 and sets $(70, 0.5, 0.75)$. Matching weights accumulate, and the last matching branch determines scalar settings before the transformations in Table 6. This fixed program adapts retrieval at successive nodes.

C.1 Definitions of the Table 3 controls

Comparison scope. The upper block uses the final two-head, 128-path SW predictor, with five-seed mean rollout scores. The lower block uses an earlier frozen single-head SW predictor: $(N, \alpha) = (40, 1)$ for PBCseq and $(10, 0.5)$ for MIMIC-IV, with single-run scores. Final SW and Original frozen SW are the reference rows for their respective blocks. Each variant is compared with its own block's reference; differences across blocks cannot be attributed to a single component. No survival heads are refitted for these controls.

Anchor only. This control directly predicts survival from the patient's landmark-visible query history using the same fitted survival heads, without retrieving transitions or simulating paths. It sets $\alpha_h = 0$ in Equation 6, giving $S_{h,0}(t) = \exp[-H_{h,z_0}(t)]$. Final anchor only averages the two query-head curves with the final model's head weights; the lower-block Anchor only uses its original single head. This comparison tests the contribution of adding trajectory simulation to direct prediction.

Geometric retrieval / Final geometric rollouts. Geometry preserves the fitted heads, path count, anchor mixture, aggregation, prognostic coordinate when present, and sampling scheme, but removes the entire retrieval program. Retrieval uses geometric distance without program-defined weights, interaction groups, or conditional actions. Its nominal k and temperature follow the stored geometric-control protocol. Thus the comparison tests the whole program, including adaptive neighborhood and temperature choices, rather than semantic distance alone. The separate No semantic distance control in Table 4 retains program actions while setting semantic mixing to zero.

One path. For the original single-head model, this variant sets $N = 1$ and uses the first path from the same saved trajectory bank. The head, retrieval program, anchor mixture α, and pooling rule remain fixed. It tests the contribution of combining multiple simulated futures rather than relying on a single sampled continuation.

Mean hazard aggregation. This variant reuses the original model's saved paths, head, N, and α, but replaces probability pooling ($q = 1$) with mean-hazard pooling ($q = 0$) in Equation 6. It therefore changes how the same trajectories are combined, not which trajectories are retrieved. The final predictor in the upper block already uses $q = 0$; this row is an aggregation control for the earlier model only.

Fixed template. This control replaces the selected program with a fixed, non-LLM rule template. It starts with unit feature weights, no interaction groups, $k = 50$, temperature 0.5, and semantic fraction 0.5. Two rules inspect the first eight features in the stored feature order: any missing feature multiplies those eight weights by 0.5 and sets $(k, \tau) = (100, 1)$; any feature in the high-level category multiplies them by 1.5 and sets $(k, \tau) = (25, 0.25)$. Matching rules execute in that order. The head, N, and α match the original SW, but the template has its own retrieval defaults. This is a fixed-rule alternative, not a strict single-factor ablation of program generation.

Permuted program. A seeded one-to-one permutation reassigns feature names consistently in the program's base weights, interaction groups, branch conditions, and branch weight multipliers. It preserves the branch structure, numerical rule values, and matched head, N, α, and retrieval settings. Patient feature values are not shuffled. This control tests whether the program benefits from associating its rules with the intended clinical features, beyond retaining the same program structure.

No dynamic branches. This variant removes all conditional branches while retaining the program's base feature weights, interaction groups, default action, and fallback rule. The original head, path count, anchor mixture, and numerical retrieval configuration remain fixed, and local inference is rerun

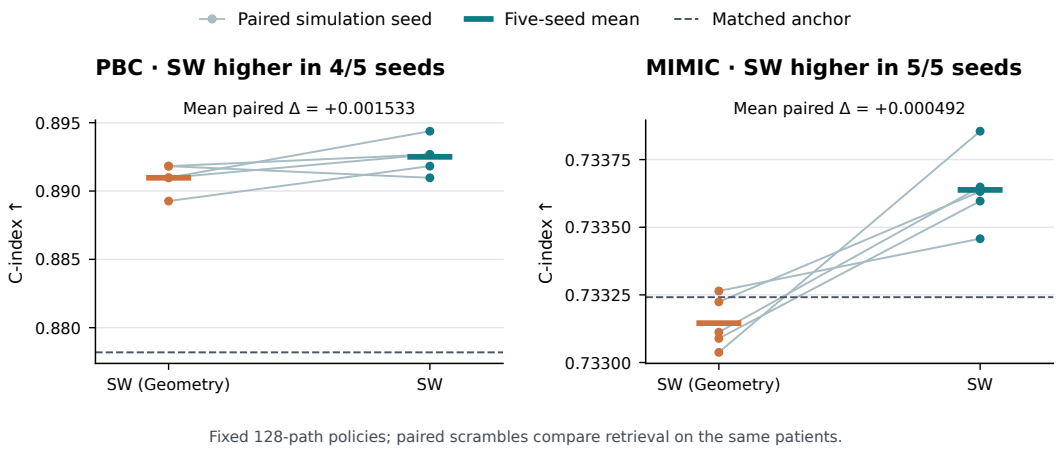


Figure 4: **Fixed-policy simulation stability.** Paired points compare SW and geometric retrieval under identical fresh simulation seeds. The PBCseq and MIMIC-IV representatives use 128 paths and shared fitted heads. Five paired scrambles measure simulation variability on the same patients.

with common random seeds. It tests state-dependent branch execution; when the retained fallback requests geometry in the absence of a matching branch, that fallback still applies.

No feature reweighting. This variant sets every base feature weight to one and removes all branch-specific weight multipliers. It retains branch conditions and their actions on neighborhood size, temperature, and semantic fraction, together with interaction groups and missingness handling. The original head and numerical configuration remain fixed, and local inference is rerun with common random seeds. It tests feature reweighting specifically, rather than removing all semantic retrieval or all dynamic behavior.

D Calibration and Prefix-Training Controls

CAL-fitted probability calibration. For each representative configuration we select a single exponent γ by minimizing CAL IBS over $\{1\} \cup \mathrm{geomspace}(0.05, 4, 81)$, breaking ties towards one. The fitted value is then applied unchanged to all five saved evaluation-seed curves. CAL also supports head early stopping; the shared selection protocol is specified in Section 4.2.

Patient-balanced prefix supervision. Each TRAIN patient contributes two rows: either two copies of the query or the query plus one deterministically sampled observed TRAIN prefix. The latter can occur after the original query landmark when such a training observation exists. CAL and TEST queries remain unchanged. Two fixed head configurations are refitted per variant, and baseline cumulative hazards are re-estimated on the original TRAIN query risk set. This comparison isolates the training intervention within its own fixed retrieval setup; the prefix setup is not the representative MIMIC-IV program described in Appendix C.

The prefix-trained anchor improves C-index over the duplicated-query anchor by 0.005839; a 500-resample paired-patient bootstrap gives a conditional 95% interval of $[0.002853, 0.009025]$. The interval is conditional on the chosen models. The anchor exceeds its corresponding SW model on all three reported metrics, so this gain is attributable primarily to training. For PBCseq, prefix supervision decreases anchor C-index from 0.881601 to 0.877342; SW changes from 0.884157 to 0.885009, while the prefix Geometry control reaches 0.886542. These results do not establish a consistent prefix benefit across cohorts. Any diagnostic that first averages prediction curves across seeds and then scores them is a separate ensemble, and is distinguished from the mean of seed-specific scores used here.

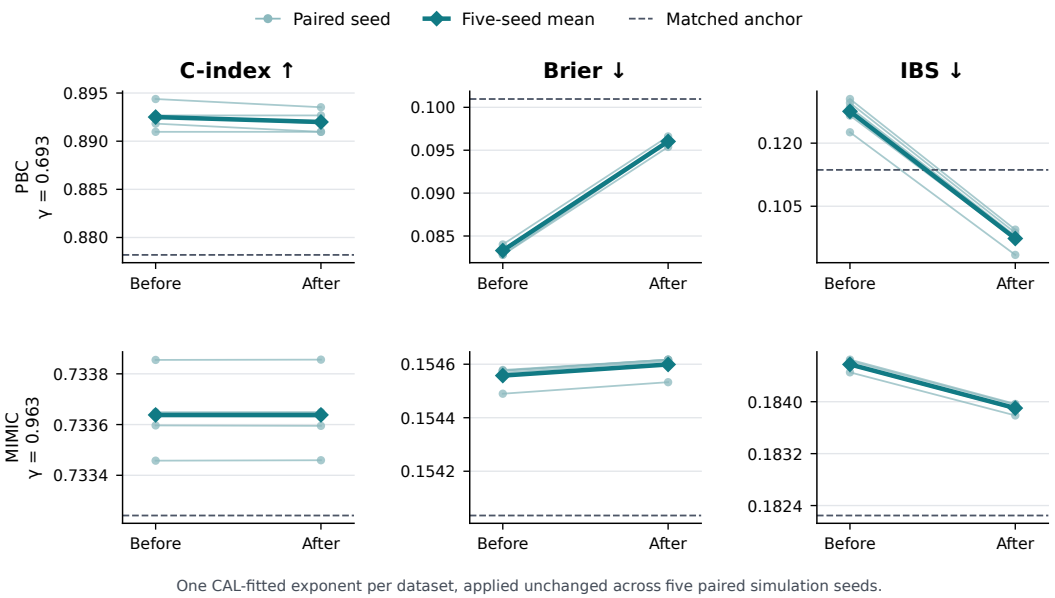


Figure 5: Paired calibration changes for the representative predictors. One CAL-fitted exponent is shared across the five evaluation scrambles of each dataset. All metrics in each condition are computed from the same saved survival curves.

E Scope of Illustrative Scenario Exploration

The illustration uses a separately frozen $q = 1$ predictor and its original 160-path bank, rather than the representative $q = 0$ configurations. For each fixed query, paths are ranked by predicted survival area on the study grid; the top and bottom 40 define the displayed scenario groups. One example per cohort is selected near the median within-query interquartile area spread without consulting outcomes. The figure reports the anchor, all-path mean, and quartile means. These are model-generated curves, not confidence intervals or treatment effects. Patient row indices remain in restricted records.

For node profiles, first targets must be observed and reached before the final prediction horizon. A feature contributes only when at least five targets per quartile have observed operands. Upper-minus-lower standardized contrasts are computed within each query and then averaged: 47–63 PBCseq queries contribute depending on the feature, and all 2,713 MIMIC-IV queries contribute. Intermediate covariates, times, and donor composition vary. Because grouping uses predicted area itself, curve separation describes scenario spread; observed-edge provenance makes hypotheses traceable without identifying intervention effects.

F Reproducibility and Artifact Scope

DeepSurv, single-risk DeepHit, and neural MTLR use pycox implementations (Kvamme et al., 2019; Fotso, 2018) with two-layer MLPs, batch normalization, dropout 0.1, and Adam optimization. Training uses batch size 64 and at most 100 epochs, with CAL-loss early stopping (patience 12). Preprocessing is fitted on TRAIN only; DeepHit and MTLR use 30 duration bins. DeepSurv uses a mini-batch Cox loss with Breslow survival estimation; DeepHit uses $\alpha = 0.2$ and $\sigma = 0.1$.

F.1 LLM configuration and prompts

Model and generation settings. We use DeepSeek-V4-Flash with temperature 0.6, a maximum output of 12,000 tokens, and JSON-object output. Each request asks for eight candidate programs, which are validated before evaluation.

System prompt (complete). "You design survival retrieval policies. Return one JSON object only, exactly following the DSL. No patient records or future values."

User prompt (excerpt). For brevity, we show only the core task instruction below. The full user prompt additionally contains dataset-specific feature descriptions, an algorithm summary, round-specific aggregate feedback, the DSL specification, a valid output example, and supplementary formatting instructions. Patient-level records are not included. "Return exactly 8 distinct, observation-aware executable retrieval programs in {`programs:[...]`}. Vary branch structure and clinical feature groups, not only temperatures. Include missingness/freshness/reuse branches plus level/trend branches. After three rounds without gain change groups and branch structure. Unknown provenance remains unknown. Do not infer absent measurement timestamps."