LTTS: A PREDICTOR–CORRECTOR FOR LONG-HORIZON COUNTERFACTUAL OUTCOME PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Comparing treatment plans requires predicting how outcomes evolve beyond a common observed history. Recursive models construct these trajectories by using generated states as inputs to later predictions. Without new measurements, errors in these inputs can affect the remaining continuation. To address this problem, we propose Long-term Trajectory Stabilization (LTTS), a predictor–corrector for long-horizon outcome prediction under sustained treatment and control from the same observed prefix. We use the predictor (TSA) to advance each treatment-specific trajectory, while the adaptation module (TAM) uses historical reconstruction to adjust selected generated states before they are reused. Using a frozen monitor, we trigger limited private adaptation on the current generated block, retaining the shared reconstruction modules without using new outcome labels. The updated window supports the next prediction; prescribed treatments and reported outcomes are preserved. Across two synthetic benchmarks and a MIMIC-IV-derived Cox model-recovery benchmark, we obtain 4.2–32.6% lower mean NMAE than the best evaluated external baseline per setting in 150-step evaluations over 30 seeds. With the predictor held fixed, we observe lower mean outcome error with feedback. Subsequent predictions also improve after a single correction when further corrections are disabled. Together, these results support combining trajectory prediction with selective state feedback to improve long-horizon outcome prediction under the evaluated treatment continuations.

1 INTRODUCTION

Once a treatment regimen is selected, it may remain in place until a later assessment. Forecasts of how outcomes may evolve during this period can provide a reference for monitoring and possible later intervention. For this setting, we predict outcome trajectories under sustained treatment and control from the same observed history (Hernán et al., 2000; Robins, 1986; Xiong et al., 2024). For this prediction task, longitudinal causal methods provide a basis under explicit assumptions, while neural sequence models offer flexible representations of outcome evolution (Robins et al., 2000; Robins, 1994; Bica et al., 2020b; Li et al., 2020).

To generate these forecasts, recursive models advance one step at a time. Alongside the outcome, they generate states that become inputs to later predictions. Each state therefore has two roles: it describes the current step and helps determine the remaining continuation. Once the observed prefix ends, neither treatment branch has new measurements with which to replace its generated inputs. A small intermediate error can consequently influence many later outcomes (Janner et al., 2019; Moerland et al., 2023). Long-horizon prediction depends both on the predictions made at each step and on how the generated states are used to continue the trajectory.

To manage these recursive inputs, we combine trajectory prediction with selective feedback in Long-term Trajectory Stabilization (LTTS). We use two Temporal Stability Architecture (TSA) branches to advance treatment and control trajectories. The Temporal Adaptation Module (TAM) learns to reconstruct historical window blocks and selectively adjusts generated inputs before the next TSA step (Figure 1). When triggered, TAM adapts its reconstruction to the current generated block. Feedback changes selected inputs for future predictions while preserving treatments, static attributes, the observed prefix, and reported outcomes. We evaluate the complete procedure using subsequent outcome prediction error.

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.

We obtain lower mean NMAE than the best evaluated external baseline with LTTS in each setting (Table 2). Using the same TSA checkpoints, we observe lower errors with feedback at early, middle, and late prediction depths (Figure 2). Feedback also lowers mean outcome error when we train the outcome head on predicted states or replace TSA with TimeKAN or TimeMixer (Tables 3a and 10). Full LTTS yields lower mean error than the tested calibrated frozen reconstruction and fixed-rule controls (Tables 9a and 8). After a single correction, it also achieves lower subsequent mean error than calibrated frozen feedback, with further TAM disabled (Table 3b).

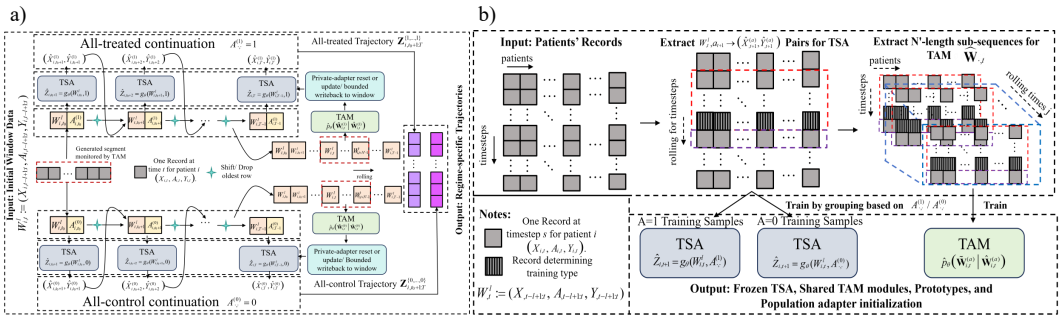


Figure 1: **LTTS overview.** (a) Independent TSA branches produce predictions for the all-one and all-zero continuations; each branch's gated TAM session can update a private adapter and write selected reconstructed entries into a later operational window. (b) We organize patient windows into continuation-grouped TSA inputs and TAM blocks used to train the offline TAM modules and prototype memory.

Our contributions are:

- We develop LTTS for two-arm counterfactual outcome prediction without future measurements. TSA advances the trajectories, while TAM uses historical reconstruction and limited adaptation to adjust subsequent inputs, protecting treatments, the observed prefix, and reported outcomes.
- We observe lower mean outcome error for the complete method on the studied benchmarks and isolate the contribution of state feedback using matched predictors and shared starting states. In a conditional analysis, we describe how feedback changes the input window and affects the propagation of subsequent prediction errors.

2 Related Work

Classical Causal Inference for Time-Varying Treatments. Marginal structural models, the g-formula, and structural nested mean models address longitudinal treatment effects (Robins et al., 2000; Robins, 1986; 1994). Doubly robust and targeted estimators combine treatment and outcome models; Q-learning and A-learning estimate dynamic treatment regimes (Bang & Robins, 2005; van der Laan & Gruber, 2012; Murphy, 2003; Schulte et al., 2014). With LTTS, we construct outcome trajectories under specified continuations through recursive prediction and state feedback.

Deep Models for Time-Varying Counterfactual Prediction. Representation balancing and latent-variable treatment models (Shalit et al., 2017; Louizos et al., 2017), recurrent RMSN and CRN (Lim, 2018; Bica et al., 2020b), and sequential latent-variable and autoencoder methods (Bica et al., 2020a; Kuzmanovic et al., 2021; Hatt & Feuerriegel, 2024) offer flexible models of outcomes and confounding. Neural g-computation, transformers, and continuous-time models generate forward trajectories (Rein et al., 2024; Li et al., 2020; Xiong et al., 2024; Melnychuk et al., 2022; Seedat et al., 2022; De Brouwer et al., 2022). TAFAS (Kim et al., 2025) and COSA (Im & Kwon, 2026) adapt using subsequently revealed observations or targets; COSA also uses recent context statistics. For the continuations studied here, future observations are unavailable; we therefore form LTTS feedback from the generated windows.

Learned Dynamics and Policy Evaluation. Model-based reinforcement learning learns dynamics for planning and policy optimization (Moerland et al., 2023). PETS models dynamics uncertainty, whereas MBPO branches short model rollouts from observed states and MOPO penalizes rewards

using dynamics uncertainty (Chua et al., 2018; Janner et al., 2019; Yu et al., 2020). GALILEO addresses behavior-policy selection bias in counterfactual models, while causal policy evaluation studies the effects of policy changes (Chen et al., 2023; Athey, 2015). With LTTS, we evaluate externally specified continuations with a fixed predictor and adaptive feedback to its future inputs.

Table 1: **Mechanistic positioning of related methods.** The rows identify the intervention stage and modeled object of each method.

Method	Stage	Mechanism or object
Scheduled Sampling (Bengio et al., 2015)	Training curriculum	Gradually replaces true previous tokens with generated previous tokens.
Professor Forcing (Lamb et al., 2016)	Adversarial training	Encourages teacher-forced and free-running recurrent dynamics to align.
DAgger (Ross et al., 2011)	Iterative imitation learning	Aggregates expert-supervised labels on observations induced by the learner.
MBPO (Janner et al., 2019)	Policy training	Branches short model rollouts from real data to limit model-bias exposure.
MOPO (Yu et al., 2020)	Offline policy optimization	Penalizes rewards with an uncertainty term from the learned dynamics.
LTTS	Inference-time rollout	Gated adaptation corrects future inputs while preserving reported predictions.

3 Problem Setup

For N patients at times $t = 1, \ldots, T$, we observe covariates $X_{i,t} \in \mathbb{R}^p$, binary intervention $A_{i,t} \in \{0, 1\}$, and outcome $Y_{i,t} \in \mathbb{R}$. We write $Z_{i,t} := (X_{i,t}, Y_{i,t})$ and the completed history $H_{i,1:t} := (\bar{X}_{i,1:t}, \bar{A}_{i,1:t}, \bar{Y}_{i,1:t})$, where bars collect the respective sequences through t and H includes the observed patient-level static attributes used by the assignment process.

Objective. We fix a decision time t_0 after observing $H_{i,1:t_0} = h_{i,1:t_0}$. For any $t > t_0$, we let $\bar{a}_{t_0+1:t} \in \{0, 1\}^{t-t_0}$ denote a prespecified binary intervention continuation. For $s \in \{t_0 + 1, \ldots, t\}$, we let $Z^{\bar{a}}_{i,s}$ denote the potential state at time s under the realized intervention prefix $\bar{A}_{i,1:t_0}$, followed by the specified continuation $\bar{a}_{t_0+1:s}$. The joint potential trajectory defines the recursive state interface:

$$\mathbf{Z}^{\bar{a}}_{i,t_0+1:t} = \left(Z^{\bar{a}}_{i,t_0+1}, \ldots, Z^{\bar{a}}_{i,t}\right). \tag{1}$$

With LTTS, we recursively produce a deterministic point prediction $\widehat{\mathbf{Z}}^{\bar{a}}_{i,t_0+1:t}$ for each specified continuation. We primarily evaluate its outcome component Y; the covariates provide auxiliary states for recursive prediction. Action dispatch also defines rollouts for changing binary continuations. Our primary evaluation uses the all-control continuation $\bar{a} = \mathbf{0}$ and the all-treated continuation $\bar{a} = \mathbf{1}$.

Temporal Convention and Reference Trajectories. For each time s, $H_{i,1:s-1}$ denotes the completed patient history available immediately before intervention $A_{i,s}$. In LTTS, we impose $A_{i,s}$ before the state $Z_{i,s}$ is realized. For the synthetic observed prefix $s \leq t_0$, we assign treatment from static attributes, lagged states and outcomes, and assignment noise. After t_0, each branch follows the externally prescribed continuation $\bar{a}$, and the benchmark generator's transition equations define the corresponding reference trajectory $Z^{\bar{a}}_{i,s}$. Using the finite observed prefix, we apply LTTS to recursively predict this trajectory from its finite operational window W^l.

Causal interpretation and benchmark evaluation. For an observational causal interpretation, we state the standard longitudinal conditions (Robins, 1986; 1994; Robins et al., 2000). For each $s > t_0$, with $H_{i,1:s-1}$ the completed history immediately before $A_{i,s}$ (including static attributes), the conditions are: consistency, $Z_{i,s} = Z^{(\bar{A}_{i,1:s})}_{i,s}$; positivity, $\mathbb{P}(A_{i,s} = a \mid H_{i,1:s-1} = h) > 0$ for $a \in \{0, 1\}$ and $h \in \mathrm{supp}(H_{i,1:s-1})$; and sequential exchangeability, $\{Z^{\bar{a}}_{i,u} : u = s, \ldots, T\} \perp A_{i,s} \mid H_{i,1:s-1}$ for each fixed continuation $\bar{a}_{s:T}$. These conditions govern identification from

observational histories. For benchmark evaluation, we compare the predicted continuations with reference trajectories supplied by the specified generators.

Predictive Target and Approximation. Under a fixed prescribed continuation, the benchmark generator supplies a reference law for the trajectory conditional on the observed prefix. With LTTS, we approximate the corresponding target path with deterministic predictions from a finite window. To describe the resulting approximation, we fix the continuation and suppress its superscript. We define $m_s := \mathbb{E}[Z_s \mid H_{1:s-1}]$ and $m_s^{(l)} := \mathbb{E}[Z_s \mid W_{s-1}^l]$ under this fixed-regime reference law. Assuming these conditional means exist, the triangle inequality gives

$$\underbrace{\|g_\theta(W_{s-1}^l, a_s) - Z_s\|}_{\epsilon_s} \le \underbrace{\|g_\theta(W_{s-1}^l, a_s) - m_s^{(l)}\|}_{\epsilon_s^{\text{fit}}} + \underbrace{\|m_s^{(l)} - m_s\|}_{\eta_s^{(l)}} + \underbrace{\|m_s - Z_s\|}_{\nu_s}. \tag{2}$$

The first term measures error relative to the regime-specific window regression, reflecting fitting accuracy as well as any mismatch between observational training and the reference law. The remaining terms describe information lost by truncating history and unpredictable innovation. In Appendix B.1, we analyze propagation assuming a bound on the reference-window error along the trajectory.

4 Architecture

To construct these trajectories, we alternate prediction with selective state feedback in LTTS (Figure 1). TSA generates the next state and outcome; TAM adjusts allowed entries in the generated window before the next prediction. Their repeated interaction constructs each treatment continuation.

4.1 Operational Framework & Notation

We use the operational window $W_{i,t}^l := (X_{i,t-l+1:t}, A_{i,t-l+1:t}, Y_{i,t-l+1:t}) \in \mathcal{W}$ to represent history through the l most recent observed or generated records, including earlier TAM writeback. The indexed time T is supplied and advances deterministically; it is not generated by TSA, and reference and predicted paths use the same index.

For window-based prediction, we dispatch two independent TSA branches by $g_\theta(W, a) := g_{\theta_a}(W)$, where $\theta = (\theta_0, \theta_1)$. On synthetic data the branches carry $X_{1:3}$, predict $X_{4:5}$ and Y, and do not receive hidden X_6; on MIMIC they carry patient PCA coordinates and assemble deterministic time and lag-risk slots with the outcome (details are in Appendix D). We record the TSA emission before TAM correction.

The shift operation then appends this emission to the next length-l window. A buffer of N' complete, potentially overlapping windows forms $\mathbf{W}_{i,t} \in \mathbb{R}^{N' \times l \times d}$, with $d = \dim(X) + 2$ and $D_{\text{in}} = N'ld$. We use the frozen offline TAM to produce $\widetilde{\mathbf{W}}_{i,t}^{\text{gate}}$ and measure its average squared discrepancy; a separate private-adapter branch may produce a proposal after a trigger. We restrict changes to later operational inputs, protecting reported emissions, the observed prefix, treatment, static variables, and reported outcomes.

4.2 Temporal Stability Architecture (TSA)

TSA advances the trajectory under the selected action. We assign training windows to branches by the treatment at the target time, retaining their factual treatment histories (Algorithm 1). For each branch, we use a residual CNN for local patterns, a 64-unit LSTM for temporal information, four-head self-attention for history aggregation, and an MLP mapping head (He et al., 2016; LeCun et al., 1998; Hochreiter & Schmidhuber, 1997; Vaswani et al., 2017). The resulting representation feeds a two-dimensional state head and an outcome head whose 67 inputs concatenate the representation, state, and treatment. We train the state head with MSE; the outcome head receives the true next state during training and the predicted synthetic state or deterministic MIMIC slots at inference. Layer widths, optimization, regularization, and final-weight rules are specified in Appendix E; Figure 6(a) summarizes the architecture.

4.3 TEMPORAL ADAPTATION MODULE (TAM)

We use historical reconstruction to draw on relationships among variables and time steps learned from observed histories. When these relationships remain informative under the prescribed continuation, reconstruction may reduce deviations introduced by recursive prediction before they propagate to later outcomes.

In TAM, we use training-derived prototypes to reconstruct complete blocks of $N' = 10$ contiguous monitoring windows, with shape $N' \times l \times d$ and flattened dimension $D_{\text{in}} = N'ld$. A shared encoder feeds prototype retrieval and fusion, followed serially by the private adapter and decoder to reconstruct the block. During offline training, we update the full network using normalized reconstruction MSE and kernel-L_2 coefficient 10^{-5}. We use validation-loss-selected weights to initialize both the fully frozen monitor and the adaptation branch. When triggered, we update the latter's private adapter for five online steps on the current generated block, adapting the reconstruction mapping while the monitor and shared modules remain frozen. Layer and training details are in Appendix E, with the reconstruction architecture shown in Figure 6(b).

For an eligible complete block, we denote the frozen monitor reconstruction by $\widetilde{\mathbf{W}}_{i,t}^{\text{gate}}$ and define $d_{i,t} := D_{\text{in}}^{-1} \| \operatorname{flat}(\widetilde{\mathbf{W}}_{i,t}^{\text{gate}}) - \operatorname{flat}(\mathbf{W}_{i,t}) \|_2^2$. The strict trigger is $d_{i,t} > \delta_{\text{corr}}$, where $\delta_{\text{corr}} = \mu_{\text{val}} + 3s_{\text{val}}$ is calibrated on validation block discrepancies using the sample standard deviation. We keep the monitor unchanged during private adaptation and hold the window block fixed during the five private updates. The adapted private state persists across triggers within each patient–continuation call.

Selective Writeback. For writeback, we let $\widetilde{W}_{i,t}^{\text{last}}$ be the last reconstructed full window in $\widetilde{\mathbf{W}}_{i,t}^{\text{prop}}$ and let $\widetilde{R}_{i,t}^{\text{last}}$ be its last generated row. We define $P_{i,t} := \operatorname{Lift}_l(\widetilde{R}_{i,t}^{\text{last}}, \overline{W}_{i,t}^{l})$ as the same-shape proposal obtained by replacing only that last generated row. The binary mask $M_{i,t}$ is zero outside that row and protects static variables, treatment, the observed prefix, and the already reported outcome. We use it to select $X_{4:5}$ on synthetic data and only the MIMIC lag-risk slot x_5 on the MIMIC-derived path; deterministic x_4 is never a TAM writeback coordinate. At eligible steps, with $\gamma_{i,t} = \mathbf{1}\{d_{i,t} > \delta_{\text{corr}}\}$,

$$W_{i,t}^{l} = \overline{W}_{i,t}^{l} + \gamma_{i,t} M_{i,t} \odot (P_{i,t} - \overline{W}_{i,t}^{l}), \tag{3}$$

where $\odot$ denotes entrywise multiplication. Otherwise, we retain $W_{i,t}^{l} = \overline{W}_{i,t}^{l}$, and no proposal is needed. Writeback follows the triggered private-adapter update in Full LTTS. We evaluate this feedback design through later outcome error, using calibrated frozen reconstruction and a single correction (Table 3; calibration and local protocols in Appendix I.2).

4.4 RECURSIVE PREDICTION PROTOCOL

For each continuation, we dispatch its independently trained branch and initialize W_{i,t_0}^{l}, a fresh adaptation private state from restored offline TAM weights, and an empty buffer. After each TSA emission, we shift the window and append the resulting window to the buffer. Before N' windows accumulate, we use the provisional window unchanged. At each complete block, we evaluate the frozen monitor; when its discrepancy exceeds δ_{corr}, we take five steps in the separate private branch and apply its proposal by Equation (3). We then clear the buffer, and the recorded emissions form $\widehat{\mathbf{Z}}_{i,t_0+1:T}^{\bar{a}} = (\widehat{Z}_{i,t_0+1}, \ldots, \widehat{Z}_{i,T})$. We evaluate this procedure separately for all-treated $\bar{a}_{t_0+1:T}^{(1)} = (1, \ldots, 1)$ and all-control $\bar{a}_{t_0+1:T}^{(0)} = (0, \ldots, 0)$. We give the full algorithms in Appendix E.

Error Propagation. To describe writeback in the window-only recursion, we let $E_h = \|\widehat{W}_h - W_h\|$ denote the operational-window deviation, $e_{h+1} = \|\widehat{Z}_{h+1} - Z_{h+1}\|$ the recorded state-prediction error, and $\Delta_h = \overline{E}_{h+1} - E_{h+1}$ the signed effect of writeback, where $\overline{E}_{h+1}$ is the provisional deviation before correction. Here $h = t - t_0$ is prediction depth, L is the local Lipschitz constant, and ϵ_{h+1} is the one-step error evaluated on the reference window. Under the conditional local-Lipschitz and bounded reference-window assumptions in Appendix B.1, the recursion gives

$$e_{h+1} \le L E_h + \epsilon_{h+1}, \qquad E_{h+1} \le (1 + L) E_h + \epsilon_{h+1} - \Delta_h.$$

Positive Δ_h reduces operational-window deviation and changes the input used by later predictions. We give the derivation in Appendix B.1 for the synthetic window-based interface.

5 EXPERIMENTS

We evaluate complete LTTS trajectories under all-control and all-treated continuations from the same prefix. We compare LTTS with baseline methods to evaluate its overall predictive performance. Using matched TSA checkpoints, we measure the effect of removing feedback; additional controls examine private adaptation and frozen writeback strength.

5.1 DATA GENERATION PROCESSES

Synthetic (linear outcome) and Synthetic (nonlinear outcome) share treatment-dependent state-transition equations. For MIMIC-derived Cox recovery, we evaluate risk curves from a frozen Cox model fitted using observed patient covariates. In additional studies, we vary the prescribed treatment continuation and predictor backbone. For the main comparisons, we use $T = 200$, with a common subject-specific prefix through $t_0 = 50$ and separate target suffixes of horizon $H = 150$ under fixed all-control and all-treated continuations. Detailed data-generation processes are provided in Appendix D.

5.1.1 SYNTHETIC ENVIRONMENTS: LINEAR AND NONLINEAR OUTCOMES

For each scenario, we generate $N = 1000$ longitudinal trajectories and use an outcome-blind subject split of $640/160/200$ for training, validation, and testing. The autoregressive state equations include periodic forcing and random innovations, with history-dependent treatment assignment through $t_0 = 50$. The unobserved state X_6 influences both treatment assignment and outcomes. We use heterogeneous linear effects with a time trend and a shared multilayer perceptron as the respective outcome mechanisms. Under a fixed action, accumulated dose approaches a constant, while periodic inputs and innovations remain in the state equations. For both outcome mechanisms, we clip the outputs to $[0, 100]$. For each subject, we construct all-control and all-treated branches sharing the same prefix and exogenous innovations, differing only in the imposed continuation after t_0.

5.1.2 MIMIC-DERIVED COX RECOVERY

We split the cohort of $N = 7297$ patients into $5103/1102/1092$ for training, validation, and testing. To identify the index admission, we sort patients by subject, admission time, and admission ID, and use the last hospitalization. We exclude patients who die in hospital during that admission or have deathdate no later than discharge; time zero is index discharge. Using procedure codes 9671, 9672, 5A1935Z, 5A1945Z, and 5A1955Z, we identified inpatient invasive mechanical ventilation (IMV). We record deaths within 365 days after discharge as events and administratively censor patients without a recorded event in this interval at day 365. On the training split, we fit a penalized Cox model to these observed survival pairs and freeze it; its covariates and preprocessing are detailed in Appendix D. Training-set PCA retains 24 dimensions. With this frozen Cox model, we generate the risk prefix through $t_0 = 50$ using each patient's covariates and observed IMV status; separate $A_t = 0$ and $A_t = 1$ suffixes run over $t = 51, \ldots, 200$ and are opened for paired scoring after final weights, configurations, and thresholds are frozen. For each suffix label, we set the corresponding IMV value in the frozen Cox predictor to generate that branch's risk curve. These risk curves follow the frozen Cox model's time-dependent cumulative hazard. We evaluate model recovery in this task while retaining observed patient heterogeneity.

Evaluation Metrics. For each benchmark, we let $\sigma_{Y,\mathrm{train}}$ denote the population standard deviation of the training-prefix outcomes. For each seed and prescribed continuation, we pool outcome errors over all test subjects and the complete 150-step suffix, with $\mathrm{NMAE} = \mathrm{MAE}/\sigma_{Y,\mathrm{train}}$ and $\mathrm{NRMSE} = \sqrt{\mathrm{MSE}}/\sigma_{Y,\mathrm{train}}$. We report the mean and sample standard deviation across 30 paired seeds. Because normalization is based on a standard deviation rather than a bounded range, either metric may exceed one.

5.2 EXPERIMENTAL SETUP AND BASELINES

In Table 2, we compare LTTS with seven baselines: marginal structural models (MSM), structural nested mean models (SNMM), Recurrent Marginal Structural Network (RMSN), Counterfactual

Recurrent Network (CRN), G-Net, G-Transformer, and GALILEO. We provide all methods with the same subject splits, benchmark-defined prefixes, and fixed continuations. We freeze configurations and inference-time thresholds before post-t_0 scoring. For the neural baselines, we retain method-specific architectures and use training caps of 300 epochs (per stage for RMSN) or 300 outer iterations for GALILEO. We stop after 20 consecutive validation checks without improvement in NMAE and restore the lowest-validation-NMAE checkpoint per seed. We train TAM in an additional offline stage of up to 50 epochs and perform five private updates per trigger. Appendices E–J provide algorithms, configurations, branch diagnostics, supplementary results, and reproducibility details.

5.3 Results and Analysis

Table 2: **Main results under both fixed continuations.** In each panel, we report outcome NMAE and NRMSE for the complete 150-step suffix under the indicated continuation. We compute each metric after aggregating the suffix separately for each seed, then summarize it as mean $\pm$ sample standard deviation over 30 paired seeds; lower is better.

Panel (a): all-one continuation									
Scenario	Metric	LTTS	MSM	SNMM	RMSN	CRN	G-Net	G-Transformer	GALILEO
Synthetic (linear outcome)	NMAE	0.248 ± 0.035	0.557 ± 0.121	0.528 ± 0.143	0.415 ± 0.020	0.368 ± 0.189	0.416 ± 0.251	0.386 ± 0.052	0.371 ± 0.069
	NRMSE	0.335 ± 0.027	0.641 ± 0.154	0.683 ± 0.171	0.581 ± 0.022	0.528 ± 0.179	0.509 ± 0.275	0.482 ± 0.063	0.468 ± 0.081
Synthetic (nonlinear outcome)	NMAE	0.218 ± 0.021	0.472 ± 0.064	0.407 ± 0.038	0.282 ± 0.074	0.264 ± 0.040	0.334 ± 0.120	0.332 ± 0.049	0.314 ± 0.061
	NRMSE	0.273 ± 0.028	0.662 ± 0.083	0.574 ± 0.052	0.346 ± 0.079	0.337 ± 0.053	0.438 ± 0.186	0.431 ± 0.058	0.412 ± 0.074
MIMIC-derived Cox recovery	NMAE	0.298 ± 0.120	0.501 ± 0.097	0.389 ± 0.116	0.334 ± 0.111	0.466 ± 0.131	0.395 ± 0.172	0.374 ± 0.051	0.359 ± 0.074
	NRMSE	0.454 ± 0.136	0.676 ± 0.121	0.559 ± 0.145	0.486 ± 0.139	0.643 ± 0.158	0.547 ± 0.206	0.548 ± 0.071	0.527 ± 0.096
Panel (b): all-zero continuation									
Scenario	Metric	LTTS	MSM	SNMM	RMSN	CRN	G-Net	G-Transformer	GALILEO
Synthetic (linear outcome)	NMAE	0.385 ± 0.100	0.608 ± 0.034	0.548 ± 0.117	0.484 ± 0.026	0.436 ± 0.141	0.458 ± 0.137	0.414 ± 0.047	0.402 ± 0.073
	NRMSE	0.527 ± 0.143	0.696 ± 0.046	0.684 ± 0.148	0.582 ± 0.035	0.614 ± 0.172	0.651 ± 0.169	0.566 ± 0.061	0.551 ± 0.089
Synthetic (nonlinear outcome)	NMAE	0.224 ± 0.014	0.541 ± 0.108	0.477 ± 0.028	0.645 ± 0.199	0.453 ± 0.126	0.482 ± 0.025	0.318 ± 0.046	0.296 ± 0.054
	NRMSE	0.281 ± 0.041	0.679 ± 0.139	0.576 ± 0.037	0.722 ± 0.201	0.644 ± 0.158	0.580 ± 0.034	0.424 ± 0.061	0.398 ± 0.071
MIMIC-derived Cox recovery	NMAE	0.276 ± 0.043	0.496 ± 0.099	0.397 ± 0.121	0.316 ± 0.020	0.432 ± 0.138	0.337 ± 0.147	0.352 ± 0.043	0.337 ± 0.067
	NRMSE	0.389 ± 0.160	0.680 ± 0.126	0.568 ± 0.149	0.452 ± 0.029	0.610 ± 0.170	0.459 ± 0.173	0.486 ± 0.059	0.468 ± 0.087

Comparing complete prediction systems in Table 2, we obtain the lowest mean NMAE and NRMSE with LTTS in all 12 cells. Relative to the best displayed baseline in each benchmark–arm setting, LTTS has 4.2–32.6% lower mean NMAE.

Contribution of the feedback stage. To measure the contribution of feedback, we compare Full LTTS with TSA-only, which removes TAM while retaining the same final-training-weight TSA checkpoints, test subjects, and normalization. With Full LTTS, we obtain lower mean NMAE in every early, middle, and late bin across the three benchmarks and both fixed arms. With Full LTTS, we also observe lower early-to-late error growth in four of six settings; the exceptions are linear all-zero and MIMIC all-one. The matched protocol is in Appendix C.2.

Table 3: **Adaptation under two diagnostic protocols.** Outcome NMAE (mean $\pm$ sample SD, 30 paired seeds; lower is better). In panel (a), we use PS-trained TSA checkpoints and score the full 150-step suffix. In panel (b), we use the original TSA: from a common state, we apply one correction, then disable TAM for 20 steps and score those future outcomes. Comparisons are within each panel. Frozen(α^*) uses validation-selected fixed writeback coefficients; Full(5) uses five private updates. We describe calibration and local protocols in Appendix I.2; PS branch results are in Appendix J.2.

	(a) PS TSA: full 150 steps		(b) Original TSA: next 20 steps	
Setting	Frozen(α^*)	Full(5)	Frozen(α^*)	Full(5)
Linear, $a = 0$	0.518 ± 0.069	0.389 ± 0.052	0.647 ± 0.092	0.619 ± 0.084
Linear, $a = 1$	0.476 ± 0.061	0.248 ± 0.035	0.571 ± 0.076	0.548 ± 0.068
Nonlinear, $a = 0$	0.501 ± 0.072	0.224 ± 0.026	0.688 ± 0.101	0.641 ± 0.089
Nonlinear, $a = 1$	0.458 ± 0.058	0.218 ± 0.021	0.621 ± 0.088	0.597 ± 0.081

Treatment switches and alternative predictors. In Figure 2, we evaluate feedback across prediction depths under fixed actions. To evaluate the response to an action change, we prescribe a switch at $t = 101$ in the nonlinear generator while retaining rolling history. The switch changes outcome inputs and subsequent state and dose evolution. With Full(5), we obtain lower mean error than frozen reconstruction without private updates (Frozen(0)) and TSA-only in both switch directions, over the

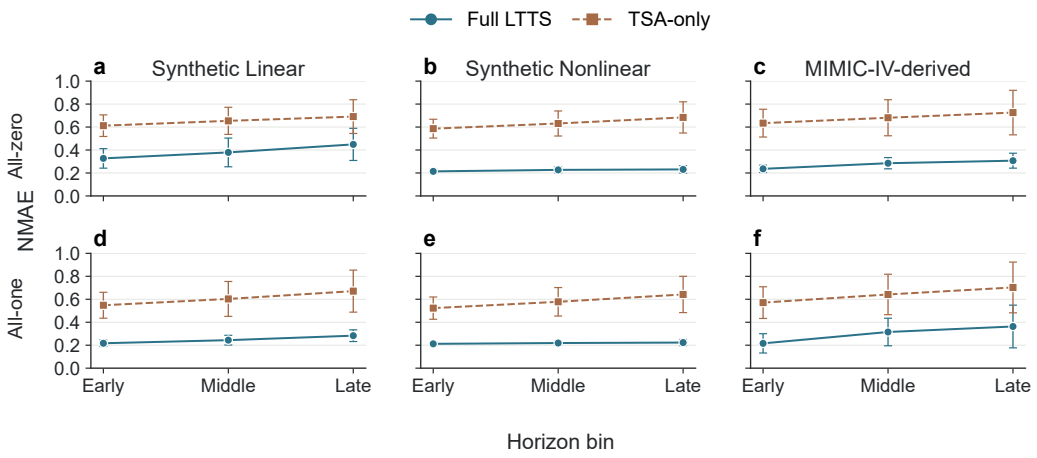


Figure 2: **Matched error across the prediction horizon.** The six panels cover the three benchmarks and two fixed continuation arms. Linear and Nonlinear label the synthetic outcome mechanisms; MIMIC labels Cox recovery. We compare Full LTTS with TSA-only using the same final TSA checkpoint, test subjects, and normalization. We report values as means $\pm$ sample standard deviation across 30 paired seeds. Early, Middle, and Late denote $h = 1$–50, 51–100, and 101–150. The matched protocol is specified in Appendix C.2.

complete suffix and the first 20 post-switch steps (Appendix I.2). When replacing the backbone with TimeKAN or TimeMixer, we also observe lower TAM-on than TAM-off means in all four reported metrics (Table 10). These experiments use the same TAM procedure with changing treatments or a different predictor.

Controls for the feedback design. For Frozen(0), we retain the offline reconstruction model, monitor, threshold, and mask, and apply its proposal with zero private updates. TAM-off/TSA-only bypasses monitoring and writeback. The Full and Frozen paths can trigger at different later steps because their writebacks change subsequent inputs.

For Frozen(α^*), we retain an input-dependent reconstruction with validation-selected mixing coefficients fixed at test time; Full(5) instead updates the private reconstruction parameters. We also examine this comparison under predicted-state (PS) training, supplying predicted next states to the outcome head while retaining factual history windows. Comparing PS TSA-only (Table 11b) and PS feedback (Table 3a) over 150 steps, we obtain lower mean NMAE with Full(5) than with PS TSA-only and calibrated Frozen in all four synthetic settings, including 24.9–55.3% lower mean NMAE than validation-calibrated Frozen(α^*). With the original TSA, we obtain 38.0–62.9% lower mean NMAE with Full(5) than calibrated Frozen over the complete 150-step suffix (Tables 2 and 9a).

As fixed-rule alternatives for the permitted feedback coordinates, we use validation-selected clipping and prototype-L_2 shrinkage with the original TSA. With Full LTTS, we obtain lower mean NMAE and NRMSE than both controls under each continuation in the equal-weight benchmark macro-average (Table 8).

Prediction after a single correction. The complete rollouts allow successive corrections to interact. To examine one correction, we start from a common state and disable further TAM for 20 steps. With adaptive writeback, we observe 3.9–6.8% lower subsequent mean NMAE than calibrated Frozen; nonlinear all-zero means are 0.641 and 0.688, respectively (Table 3b).

Component and trigger controls. We report the detailed component ablations in Appendix I.1 (Table 8), using equal-weight benchmark macro-averages over 30 paired seeds. We obtain the lowest mean in all four columns with Full LTTS; removing TAM raises all-zero NMAE from 0.295 to 0.655 and all-one NMAE from 0.255 to 0.609. The connectivity-removal rows also remain above Full

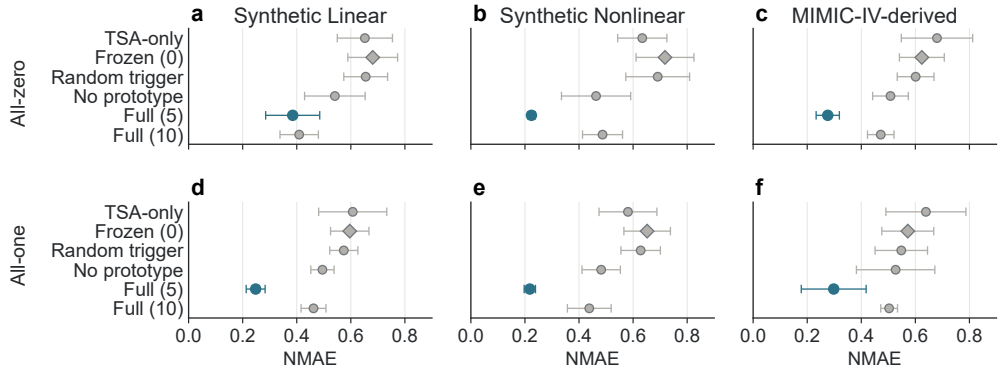


Figure 3: **Internal TAM controls.** Matched controls for three benchmarks and two fixed arms, with labels as in Figure 2. Values are means ± sample standard deviation across 30 paired seeds. For Frozen(0), we retain the frozen reconstruction writeback while disabling private updates; it is distinct from TAM-off. Random trigger and No prototype use five private updates; parentheses indicate the update count for Full TAM. Full(5) is lowest among the six displayed controls under this protocol. The matched protocol is specified in Appendix I.3.

LTTS. In Figure 3, we compare discrepancy-based and random triggering, prototype use, and update budgets; Full(5) is lowest in all six panels, and five updates yield lower mean errors than ten under the matched protocol (Appendix I.3).

In the first-trigger audit, we compare subsequent closed-loop behavior from a common state immediately before the first eligible writeback on the Full path: Full continues its TAM decisions, while TAM-off disables feedback. In this ten-seed audit, we include 29,840 eligible patient–arm–seed rollouts, each evaluated at four correlated offsets (119,360 comparisons); the pooled negative fraction is 9.4% (Figure 4; Appendix I.4).

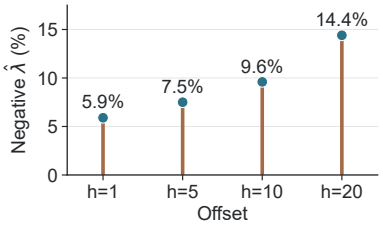


Figure 4: **Post-trigger contrasts.** Percentages of negative $\widehat{\lambda}_{\tau,h}$ contrasts among 29,840 eligible rollouts at each offset. A negative value means higher Full LTTS MSE than its matched TAM-off branch.

Computational cost. On the profiled hardware, Full(5) rollout takes 2.5–2.7× the time of Frozen(0) and 4.9–5.0× that of TSA-only (Appendix H).

6 Discussion

We find that combining TSA with selective state feedback improves long-horizon outcome prediction in the studied two-arm tasks. The matched controls support treating the handling of generated inputs as part of prediction design.

We see three potential monitoring settings: medication follow-up, rehabilitation, and equipment operation. Clinical forecasts could support follow-up under a maintenance medication regimen. Functional outcome forecasts could describe recovery during prescribed rehabilitation. In equipment monitoring, forecasts under an enabled or disabled protective mode could describe temperature or degradation trends before inspection. In each case, forecasts under the selected plan and its comparator provide a reference until the next assessment, when a new intervention may be considered.

Our evaluation covers two synthetic outcome mechanisms and fitted-Cox recovery, with selected treatment switches and predictor backbones. Further work could examine why feedback benefits vary across trajectories and evaluate the trade-off between predictive gains and adaptation cost in broader dynamic settings.

AI USE STATEMENT

We used generative AI tools to aid manuscript preparation and polish the writing, including translating and editing text, improving clarity and readability, organizing sections and tables, and assisting with reference formatting and LaTeX editing.

We reviewed and revised all AI-assisted edits to ensure that they accurately reflected the underlying research, mathematical arguments, implementation, and experimental results. We take full responsibility for the final content, claims, and artifacts in this work.

ETHICS STATEMENT

The MIMIC-IV-derived benchmark is based on deidentified, restricted-access health records. Access to these records is limited to credentialed users who complete required training and sign the applicable data use agreement (DUA), in accordance with applicable data-use requirements (Johnson et al., 2023). We collected no new data and had no direct patient interaction in this study, and we make no attempt to re-identify individuals or redistribute restricted patient-level records. The semi-synthetic Cox-generated targets are benchmark constructs intended for methodological evaluation.

REPRODUCIBILITY STATEMENT

We describe the data generation and benchmark contracts in Section 3 and Appendix D; give model training and recursive prediction procedures in Section 4 and Appendix E; state conditional theoretical assumptions and proofs in Appendix B.1; specify baselines and ablations in Appendix F through Appendix I; and provide the detailed protocol in Appendix J. In the empirical tables, we specify the dataset scale, seed counts, normalization, and aggregation, and the algorithms distinguish offline shared training from inference-time private adaptation. We provide reference LTTS training, prediction, and evaluation code, synthetic data, and baseline details with this submission.

REFERENCES

Susan Athey. Machine Learning and Causal Inference for Policy Evaluation. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pp. 5–6. ACM, Aug 2015. doi: 10.1145/2783258.2785466. URL http://dx.doi.org/10.1145/2783258.2785466.

Heejung Bang and James M. Robins. Doubly Robust Estimation in Missing Data and Causal Inference Models. *Biometrics*, 61(4):962–973, Dec 2005. ISSN 1541-0420. doi: 10.1111/j.1541-0420.2005.00377.x. URL http://dx.doi.org/10.1111/j.1541-0420.2005.00377.x.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/e995f98d56967d946471af29d7bf99f1-Paper.pdf.

Ioana Bica, Ahmed Alaa, and Mihaela Van Der Schaar. Time Series Deconfounder: Estimating Treatment Effects over Time in the Presence of Hidden Confounders. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 884–895. PMLR, 13–18 Jul 2020a. URL https://proceedings.mlr.press/v119/bica20a.html.

Ioana Bica, Ahmed M. Alaa, James Jordon, and Mihaela van der Schaar. Estimating Counterfactual Treatment Outcomes Over Time Through Adversarially Balanced Representations. In *International Conference on Learning Representations*, 2020b. URL https://iclr.cc/virtual_2020/poster_BJg866NFvB.html.

Xiong-Hui Chen, Yang Yu, Zhengmao Zhu, ZhiHua Yu, Zhenjun Chen, Chenghe Wang, Yinan Wu, Hongqiu Wu, Rong-Jun Qin, Ruijin Ding, and Fangsheng Huang. Adversarial Counterfactual Environment Model Learning. In A. Oh, T. Naumann, A. Globerson,

K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 70654–70706. Curran Associates, Inc., 2023. doi: 10.52202/075280-3097. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/df927a06a0d9f5f06d9cd4a91ce58e56-Paper-Conference.pdf.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/3de568f8597b94bda53149c7d7f5958c-Paper.pdf.

Edward De Brouwer, Javier Gonzalez, and Stephanie Hyland. Predicting the impact of treatments over time with uncertainty aware neural differential equations. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 4705–4722. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/de-brouwer22a.html.

Tobias Hatt and Stefan Feuerriegel. Sequential Deconfounding for Causal Inference with Unobserved Confounders. In Francesco Locatello and Vanessa Didelez (eds.), *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pp. 934–956. PMLR, 01–03 Apr 2024. URL https://proceedings.mlr.press/v236/hatt24a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, June 2016.

Miguel Ángel Hernán, Babette Brumback, and James M. Robins. Marginal Structural Models to Estimate the Causal Effect of Zidovudine on the Survival of HIV-Positive Men. *Epidemiology*, 11(5):561–570, Sept 2000. ISSN 1044-3983. doi: 10.1097/00001648-200009000-00012. URL http://dx.doi.org/10.1097/00001648-200009000-00012.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9 (8):1735–1780, Nov 1997. ISSN 1530-888X. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Songtao Huang, Zhen Zhao, Can Li, and LEI BAI. TimeKAN: KAN-based Frequency Decomposition Learning Architecture for Long-term Time Series Forecasting. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 28514–28529, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/46362971bfc3a97e6a271f2eb90fba17-Paper-Conference.pdf.

Jeonghwan Im and Hyuk-Yoon Kwon. COSA: Context-aware Output-Space Adapter for Test-Time Adaptation in Time Series Forecasting. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 24845–24874, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/2a8ce71baac4c89bf9ff479d8240c7d9-Paper-Conference.pdf.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to Trust Your Model: Model-Based Policy Optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5faf461eff3099671ad63c6f3f094f7f-Paper.pdf.

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):1, Jan 2023. ISSN 2052-4463. doi: 10.1038/s41597-022-01899-x. URL http://dx.doi.org/10.1038/s41597-022-01899-x.

HyunGi Kim, Siwon Kim, Jisoo Mok, and Sungroh Yoon. Battling the Non-stationarity in Time Series Forecasting via Test-time Adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(17):17868–17876, Apr 2025. ISSN 2159-5399. doi: 10.1609/aaai.v39i17.33965. URL http://dx.doi.org/10.1609/aaai.v39i17.33965.

Milan Kuzmanovic, Tobias Hatt, and Stefan Feuerriegel. Deconfounding Temporal Autoencoder: Estimating Treatment Effects over Time Using Noisy Proxies. In Subhrajit Roy, Stephen Pfohl, Emma Rocheteau, Girmaw Abebe Tadesse, Luis Oala, Fabian Falck, Yuyin Zhou, Liyue Shen, Ghada Zamzmi, Purity Mugambi, Ayah Zirikly, Matthew B. A. McDermott, and Emily Alsentzer (eds.), *Proceedings of Machine Learning for Health*, volume 158 of *Proceedings of Machine Learning Research*, pp. 143–155. PMLR, 04 Dec 2021. URL https://proceedings.mlr.press/v158/kuzmanovic21a.html.

Alex M Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. Professor Forcing: A New Algorithm for Training Recurrent Networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/16026d60ff9b54410b3435b403afd226-Paper.pdf.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. ISSN 0018-9219. doi: 10.1109/5.726791. URL http://dx.doi.org/10.1109/5.726791.

Rui Li, Zach Shahn, Jun Li, Mingyu Lu, Prithwish Chakraborty, Daby Sow, Mohamed Ghalwash, and Li wei H. Lehman. G-Net: A Deep Learning Approach to G-computation for Counterfactual Outcome Prediction Under Dynamic Treatment Regimes, 2020. URL https://arxiv.org/abs/2003.10551.

Bryan Lim. Forecasting Treatment Responses Over Time Using Recurrent Marginal Structural Networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/56e6a93212e4482d99c84a639d254b67-Paper.pdf.

Christos Louizos, Uri Shalit, Joris Mooij, David Sontag, Richard Zemel, and Max Welling. Causal Effect Inference with Deep Latent-Variable Models. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/94b5bde6de888ddf9cde6748ad2523d1-Paper.pdf.

Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Causal Transformer for Estimating Counterfactual Outcomes. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15293–15329. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/melnychuk22a.html.

Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based Reinforcement Learning: A Survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, Jan 2023. ISSN 1935-8245. doi: 10.1561/2200000086. URL http://dx.doi.org/10.1561/2200000086.

S. A. Murphy. Optimal Dynamic Treatment Regimes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 65(2):331–355, Apr 2003. ISSN 1467-9868. doi: 10.1111/1467-9868.00389. URL http://dx.doi.org/10.1111/1467-9868.00389.

Sophia M Rein, Jing Li, Miguel Hernan, and Andrew Beam. Deep Learning Methods for the Noniterative Conditional Expectation G-Formula for Causal Inference from Complex Observational Data, 2024. URL https://arxiv.org/abs/2410.21531.

James Robins. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical Modelling*, 7 (9-12):1393–1512, 1986. ISSN 0270-0255. doi: 10.1016/0270-0255(86)90088-6. URL http://dx.doi.org/10.1016/0270-0255(86)90088-6.

James M. Robins. Correcting for non-compliance in randomized trials using structural nested mean models. *Communications in Statistics - Theory and Methods*, 23(8):2379–2412, Jan 1994. ISSN 1532-415X. doi: 10.1080/03610929408831393. URL http://dx.doi.org/10.1080/03610929408831393.

James M. Robins, Miguel Ángel Hernán, and Babette Brumback. Marginal Structural Models and Causal Inference in Epidemiology. *Epidemiology*, 11(5):550–560, Sept 2000. ISSN 1044-3983. doi: 10.1097/00001648-200009000-00011. URL http://dx.doi.org/10.1097/00001648-200009000-00011.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/ross11a.html.

Phillip J. Schulte, Anastasios A. Tsiatis, Eric B. Laber, and Marie Davidian. **Q**- and **A**-Learning Methods for Estimating Optimal Dynamic Treatment Regimes. *Statistical Science*, 29(4):640–661, Nov 2014. ISSN 0883-4237. doi: 10.1214/13-sts450. URL http://dx.doi.org/10.1214/13-STS450.

Nabeel Seedat, Fergus Imrie, Alexis Bellot, Zhaozhi Qian, and Mihaela van der Schaar. Continuous-Time Modeling of Counterfactual Outcomes Using Neural Controlled Differential Equations. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 19497–19521. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/seedat22b.html.

Uri Shalit, Fredrik D. Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3076–3085. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/shalit17a.html.

Mark J. van der Laan and Susan Gruber. Targeted Minimum Loss Based Estimation of Causal Effects of Multiple Time Point Interventions. *The International Journal of Biostatistics*, 8(1), May 2012. ISSN 1557-4679. doi: 10.1515/1557-4679.1370. URL http://dx.doi.org/10.1515/1557-4679.1370.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Zhang, and JUN ZHOU. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 38626–38652, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/a7ac8a21e5a27e7ab31a5f42a0117bdb-Paper-Conference.pdf.

Hong Xiong, Feng Wu, Leon Deng, Megan Su, and Li-wei H. Lehman. G-Transformer: Counterfactual Outcome Prediction under Dynamic and Time-varying Treatment Regimes. In Kaivalya Deshpande, Madalina Fiterau, Shalmali Joshi, Zachary Lipton, Rajesh Ranganath, and Iñigo

Urteaga (eds.), *Proceedings of the 9th Machine Learning for Healthcare Conference*, volume 252 of *Proceedings of Machine Learning Research*. PMLR, 16–17 Aug 2024. URL https://proceedings.mlr.press/v252/xiong24a.html.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based Offline Policy Optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 14129–14142. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/a322852ce0df73e204b7e67cbbef0d0a-Paper.pdf.

A NOTATION AND TARGET CONTRACT

The method and theory allow any prespecified binary intervention continuation $\bar{a} = (a_{t_0+1}, \ldots, a_T)$. In the primary benchmark comparison, we instantiate the all-zero continuation $\bar{a}^{(0)} = (0, \ldots, 0)$ and the all-one continuation $\bar{a}^{(1)} = (1, \ldots, 1)$; the additional diagnostics use the stated single-correction interventions and one prespecified nonlinear switch in each direction. For unit i, t_0 is the decision time and $H_{i,1:t_0}$ is the observed history through that time. With $Z_{i,t} := (X_{i,t}, Y_{i,t})$, the regime-specific target is

$$\mathbf{Z}^{\bar{a}}_{i,t_0+1:T} = (Z^{\bar{a}}_{i,t_0+1}, \ldots, Z^{\bar{a}}_{i,T}),$$

and the primary point prediction is $\widehat{\mathbf{Z}}^{\bar{a}}_{i,t_0+1:T}$.

At prediction depth $h = t - t_0$, $W^{\bar{a}}_h$ is the reference window constructed from the target trajectory, while $\widehat{W}^{\bar{a}}_h$ is the operational window supplied to TSA after any earlier TAM writeback. The operational window is the finite history representation defined in Section 4.1; Equation 2 accounts for history truncation. Before writeback, TSA produces the raw next-state emission $\widehat{Z}_{i,t+1} = g_\theta(\widehat{W}^{\bar{a}}_h, a_{h+1})$, which is recorded as the reported prediction. The Shift operator then forms the provisional window

$$\overline{W}_{h+1} := \mathrm{Shift}(\widehat{W}^{\bar{a}}_h, (\widehat{Z}_{i,t+1}, a_{h+1})).$$

The prescribed intervention coordinate remains in every operational window. The TAM buffer contains complete generated length-l windows. After appending the current window, we denote a complete block of N' possibly overlapping windows by $\mathbf{W}_{h+1} \in \mathbb{R}^{N' \times l \times d}$ with $d = \dim(X)+2$ and $D_{\mathrm{in}} = N'ld$. The frozen offline TAM produces $\widetilde{\mathbf{W}}^{\mathrm{gate}}_{h+1}$ and the score $d_{h+1} = D_{\mathrm{in}}^{-1} \| \mathrm{flat}(\widetilde{\mathbf{W}}^{\mathrm{gate}}_{h+1}) - \mathrm{flat}(\mathbf{W}_{h+1}) \|_2^2$. If $d_{h+1} > \delta_{\mathrm{corr}}$, we update an independent private-adapter branch on the fixed block, recompute $\widetilde{\mathbf{W}}^{\mathrm{prop}}_{h+1}$, take its last reconstructed window and last generated row, and apply the selective writeback in Equation (3) to produce $\widehat{W}_{h+1}$. The frozen monitor is unchanged and the provisional window is used unchanged when the threshold is not exceeded.

For analysis, we suppress i and $\bar{a}$ when unambiguous. The operational-window deviation is $E_h = \|\widehat{W}_h - W_h\|$, the provisional deviation is $\overline{E}_{h+1} = \|\overline{W}_{h+1} - W_{h+1}\|$, and the reference-window one-step approximation error is $\epsilon_{h+1} = \|g_\theta(W_h, a_{h+1}) - Z_{h+1}\|$. With $q = 1 + L$, the one-step signed effect is $\Delta_h = \overline{E}_{h+1} - E_{h+1}$.

Table 4: Notation for the target and recursive prediction procedure.

Symbol(s)	Meaning
i, t_0, t, h, T, H	Unit, decision time, future time, prediction depth, final time index, and suffix horizon ($H = T - t_0$).
$X_{i,t}, A_{i,t}, Y_{i,t}, Z_{i,t}, H_{i,1:t_0}$	Observed covariates, intervention, outcome, joint state, and observed history through the decision time.
$\bar{a}, \bar{a}^{(0)}, \bar{a}^{(1)}, a_{h+1}$	Prespecified binary continuation, the primary all-zero and all-one continuations, and the intervention at prediction step $h+1$.
$\mathbf{Z}^{\bar{a}}_{i,t_0+1:T}, \widehat{\mathbf{Z}}^{\bar{a}}_{i,t_0+1:T}$	Regime-specific target trajectory and deterministic predicted point trajectory.
$W^{\bar{a}}_h, \widehat{W}^{\bar{a}}_h, \overline{W}_{h+1}, \mathcal{W}$	Reference window, operational window supplied to TSA, provisional shifted window, and valid-window space.
$\mathbf{W}_h, \widetilde{\mathbf{W}}^{\mathrm{gate}}_h, \widetilde{\mathbf{W}}^{\mathrm{prop}}_h$	$N' \times l \times d$ complete-window TAM block, frozen-monitor reconstruction, and adapted block proposal for writeback.
$l, N', P_{i,t}, M_{i,t}, \gamma_{i,t}$	Window length, monitoring-window cadence, aligned proposal window, protected selective-writeback mask, and gate indicator.
$g_\theta, \mathrm{Shift}, \widehat{Z}_{i,t}$	Action-dispatched TSA transition, window-shift operator, and raw TSA emission recorded as the reported prediction.
$d_h, \delta_{\mathrm{corr}}$	Full-block mean-squared reconstruction discrepancy for an eligible window block and the writeback gate threshold.
$\phi_{\mathrm{enc}}, \phi_{\mathrm{dec}}, \phi^{\mathrm{off}}_{\mathrm{priv}}$	Serial TAM encoder, decoder, and frozen offline private adapter.
$\phi_{\mathrm{adapt}}, K_{\mathrm{adapt}}$	Adaptation private adapter and its update count.
$\mathbf{Z}_{\mathrm{enc}}, \mathbf{Z}_{\mathrm{proto}}, \mathbf{M}, K, D_E$	Encoded/prototype representations, memory, count, and embedding dimension.
D_{in}	Flattened TAM block size $N'ld$.
$e_{h+1}, E_h, \overline{E}_{h+1}, \epsilon_{h+1}$	Recorded state-prediction error, operational-window deviation, provisional-window deviation, and reference-window one-step approximation error.
$m_s, m^{(l)}_s, \epsilon^{\mathrm{fit}}_s, \eta^{(l)}_s, \nu_s$	Full-history and finite-window reference means; fitting, history-truncation, and innovation terms in Equation (2).
L, q, Δ_h	Local Lipschitz constant, $q = 1 + L$, and signed one-step writeback effect.
$\|\cdot\|, \|\cdot\|^2_2$	Window/state norm and squared Euclidean reconstruction error used for the TAM block discrepancy.

B Auxiliary Error Analysis

B.1 Conditional Error-Propagation Analysis

We consider the synthetic window-based interface, where the window and prescribed action determine the next predictor input at each supplied time index. We fix a patient and continuation $\bar{a}$, suppress their indices, and let $h = t - t_0$ and $a_{h+1} = a_{t_0+h+1}$. The reference state and length-l window are Z_h and W_h; $\widehat{W}_h$ is the operational window after any writeback. TSA records $\widehat{Z}_{h+1} = g_\theta(\widehat{W}_h, a_{h+1})$ before correction and forms $\overline{W}_{h+1} = \mathrm{Shift}(\widehat{W}_h, (\widehat{Z}_{h+1}, a_{h+1}))$. Using Euclidean product norms with fixed coordinate scaling, we define

$$e_{h+1} = \|\widehat{Z}_{h+1} - Z_{h+1}\|, \quad E_h = \|\widehat{W}_h - W_h\|, \quad \overline{E}_{h+1} = \|\overline{W}_{h+1} - W_{h+1}\|.$$

Initialization from the observed prefix gives $E_0 = 0$.

Conditions. On a region containing the reference and operational windows over the analyzed horizon, we assume $\|g_\theta(W, a) - g_\theta(W', a)\| \le L\|W - W'\|$ uniformly over the prescribed actions. For two-branch dispatch, L bounds both branch constants. The reference-window approximation error is $\epsilon_{h+1} = \|g_\theta(W_h, a_{h+1}) - Z_{h+1}\| \le \bar{\epsilon}$. It includes the fitting, finite-window and innovation contributions in Equation (2). These pathwise conditions apply over the analyzed horizon.

Recursion and proof. The triangle inequality and the Lipschitz condition give

$$e_{h+1} \le \|g_\theta(\widehat{W}_h, a_{h+1}) - g_\theta(W_h, a_{h+1})\| + \epsilon_{h+1} \le LE_h + \epsilon_{h+1}.$$

The shift removes the oldest row and appends the recorded prediction. Because prescribed action coordinates agree, its exact error identity is

$$\overline{E}_{h+1}^2 = E_h^2 - \|(\widehat{W}_h - W_h)_1\|^2 + e_{h+1}^2,$$

where subscript 1 denotes the removed row. Dropping the nonnegative subtracted term yields $\overline{E}_{h+1} \le E_h + e_{h+1} \le (1 + L)E_h + \epsilon_{h+1}$. We define the signed writeback effect relative to the reference window as $\Delta_h = \overline{E}_{h+1} - E_{h+1}$. Hence

$$E_{h+1} = \overline{E}_{h+1} - \Delta_h \le (1 + L)E_h + \epsilon_{h+1} - \Delta_h.$$

Without writeback, $\Delta_h = 0$; repeated substitution gives $E_H \le \sum_{r=1}^{H}(1 + L)^{H-r}\epsilon_r$.

Interpretation. The recursion separates inherited window deviation from new reference-window error. The bound accounts for possible amplification of inherited window error. Positive Δ_h reduces window deviation; negative values increase it. Writeback changes the operational window used in subsequent predictions. The sign of Δ_h depends on the benchmark reference, while reconstruction MSE targets the generated block. We therefore evaluate subsequent outcome error separately from window deviation.

C Additional Trajectory Analyses

The all-one and all-zero panels of Table 2 provide the complete-suffix comparison; in this appendix we develop a depth-wise view of the same task.

C.1 Illustrative LTTS Prediction Examples

In Figure 5, we present LTTS cases across the three benchmarks from the observed history through $t_0 = 50$, followed by separate 150-step predictions under the two fixed continuations. We construct references using the corresponding synthetic generator or frozen Cox recovery process. Aggregate performance is reported in Table 2.

C.2 Error Across the Prediction Horizon

In Figure 2, we provide the six-panel full-scale comparison for Full LTTS and matched TSA-only. We report values as means $\pm$ sample SD over 30 paired seeds for Early ($h = 1$–50), Middle (51–100), and Late (101–150); TSA-only uses the matched Full LTTS predictor checkpoint.

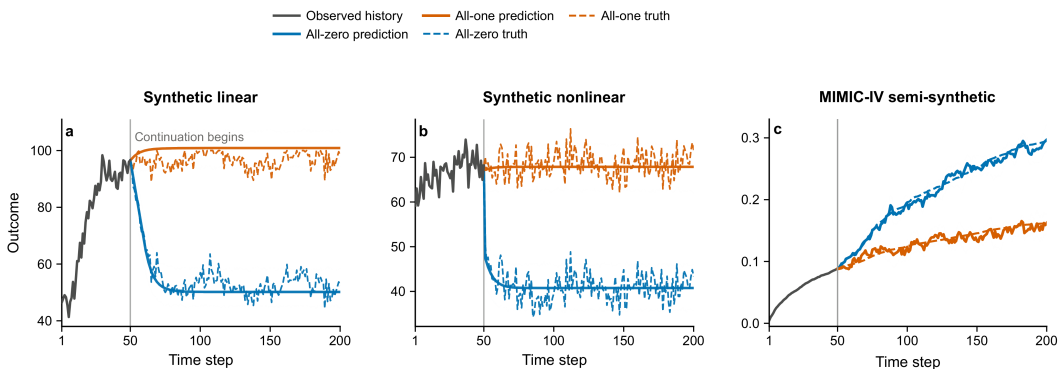


Figure 5: Illustrative LTTS prediction examples for Synthetic (linear outcome), Synthetic (nonlinear outcome), and MIMIC-derived Cox recovery. In each panel, we show the observed prefix through $t \leq 50$, followed by LTTS predictions under the all-zero and all-one continuations and the corresponding benchmark reference trajectories.

D Benchmark Construction

For the main fixed-continuation experiments, we use $T = 200$, an observed subject-specific prefix through $t_0 = 50$, and a recursive suffix of horizon $H = 150$. After t_0, we supply each branch with a fixed continuation, either $\bar{a}^{(0)} = (0, \ldots, 0)$ or $\bar{a}^{(1)} = (1, \ldots, 1)$. The primary targets are the two regime-specific future trajectories $\mathbf{Z}^{\bar{a}^{(0)}}$ and $\mathbf{Z}^{\bar{a}^{(1)}}$ generated under these continuations.

D.1 Synthetic Environments: Linear and Nonlinear Outcomes

For each scenario, we use $N = 1000$ trajectories and an outcome-blind subject split of $640/160/200$ for training, validation, and testing. The main context length is $l = 10$. The exported input coordinates are $X_1 = \text{age}$, $X_2 = C$, and $X_3 = B$; these observed static attributes are repeated along each patient's trajectory. We standardize continuous inputs using training-prefix statistics. The hidden state X_6 and dose carryover D are internal generator variables. Suppressing the subject index, we initialize the generator with $G_{\text{age}} \sim \mathcal{N}(60, 12^2)$ and $\text{age} = \text{clip}_{[18,90]}(G_{\text{age}})$, $G_C \sim \mathcal{N}(2.5, 1.5^2)$ and $C = \text{clip}_{[0,10]}(G_C)$, $B \sim \mathcal{N}(0, 1)$, $X_{4,0}, X_{5,0}, X_{6,0} \sim \mathcal{N}(0, 1)$, $D_0 = 0$, and $A_0 = 0$. For each scenario, $z^{\text{risk}} = (\text{age} - 60)/12 + 0.7(C - 2.5)/1.5 + \xi^{\text{risk}}$ with $\xi^{\text{risk}} \sim \mathcal{N}(0, 0.5^2)$. For the linear scenario, we additionally use $z^{\text{resp}} = \exp(\xi^{\text{resp}})$ with $\xi^{\text{resp}} \sim \mathcal{N}(0, 0.3^2)$, whereas $z^{\text{resp}} = 0$ in the nonlinear scenario.

The dynamic covariates and state variable evolve as

$$
\begin{aligned}
X_{4,t} &= 0.8X_{4,t-1} + 0.25A_{t-1} + 0.10D_{t-1} + 0.6\sin(2\pi t/120) + \varepsilon_{4,t},\\
X_{5,t} &= 0.6X_{5,t-1} + 0.20A_{t-1} + 0.08D_{t-1} + 0.3\sin(2\pi t/80) + \varepsilon_{5,t},\\
X_{6,t} &= 0.7X_{6,t-1} + 0.4A_{t-1} + 0.15X_{4,t} + 0.5\sin(2\pi t/60) + \varepsilon_{6,t},\\
D_t &= 0.9D_{t-1} + A_t.
\end{aligned}
$$

Here $X_{6,t}$ is an unobserved state in the generator and D_t is dose carryover. $X_{6,t}$ is not supplied to the prediction model and jointly influences the observational assignment and outcome mechanisms. For $t \leq t_0$, the observational policy assigns treatment from static attributes, lagged states and outcomes, and assignment noise:

$$
\begin{aligned}
p_t = \sigma\big(&-0.3 + 1.5(\text{age} - 60)/12 + 1.8(C - 2.5) + 1.5B\\
&+ 0.6X_{4,t-1} + 0.4X_{5,t-1} + 0.8X_{6,t-1}\\
&+ 0.5(Y_{t-1} - 50)/10 + 0.3t/T + 0.5z^{\text{risk}} + 0.2D_{t-1} + \eta_{A,t}\big),\\
A_t \sim \text{Bernoulli}\big(&\text{clip}_{[0.02,0.98]}(p_t)\big).
\end{aligned}
$$

The innovations are $\varepsilon_{4,t} \sim \mathcal{N}(0, 0.5^2)$, $\varepsilon_{5,t}, \varepsilon_{6,t} \sim \mathcal{N}(0, 0.7^2)$, $\eta_{A,t} \sim \mathcal{N}(0, 0.2^2)$, and $\eta_{Y,t} \sim \mathcal{N}(0, 2.5^2)$. The initial linear outcome is $Y_0^{\text{lin}} = \text{clip}_{[0,100]}(55 - 0.3(\text{age} - 60) - 1.2(C - 2.5) +$

$0.5B + \xi_0)$ with $\xi_0 \sim \mathcal{N}(0, 3^2)$, while $Y_0^{\mathrm{nl}} = 50$. The linear outcome mechanism is

$$\begin{aligned} Y_t^{\mathrm{lin}} = \mathrm{clip}_{[0,100]}\big(&50 + 0.02(\mathrm{age} - 60) - 0.8(C - 2.5) + 0.5B + 0.6X_{4,t} + 0.2X_{5,t} \\ &- 1.2X_{6,t} + 5t/T + 6z^{\mathrm{resp}}D_t \\ &+ 0.5A_t(C - X_{4,t}) + \eta_{Y,t}\big). \end{aligned}$$

For the nonlinear scenario, we let $\mathbf{v}_t = ((\mathrm{age} - 60)/12, (C - 2.5)/1.5, B, X_{4,t}, X_{5,t}, X_{6,t}, A_t, D_t/10)$. We use a shared generator MLP with input dimension 8, one hidden ReLU layer of width 64, and fixed weights ω in both branches: the seeded initialization uses standard deviation $\sqrt{2/8}$ for the first layer, resamples its action-input row as $2\mathcal{N}(0.5, 1)$, uses standard deviation $\sqrt{2/65}$ for the second layer, and sets $b_1 = 0$, $b_2 = 40$.

$$Y_t^{\mathrm{nl}} = \mathrm{clip}_{[0,100]}\left(\mathrm{MLP}_\omega(\mathbf{v}_t) + 15A_t + 7.5(D_t/10) + \eta_{Y,t}\right).$$

For each subject, we generate the prefix through t_0 once under the observational policy. At every $t > t_0$, we sample one tuple $(\varepsilon_{4,t}, \varepsilon_{5,t}, \varepsilon_{6,t}, \eta_{Y,t})$ and reuse it in both branches, while A_t is fixed to 0 or 1. Thus the branches share their prefix and future innovations and differ through the imposed continuation and its recursive state updates. We use 30 paired seeds for the synthetic evaluation.

D.2 MIMIC-Derived Cox Recovery

The MIMIC-IV-derived semi-synthetic benchmark contains $N = 7297$ patients with a subject-disjoint split of $5103/1102/1092$ for training, validation, and testing. The main context length is $l = 20$. We retain adults with valid admission and discharge times, excluding discharge before admission, in-hospital deaths at the index admission, and death on or before the discharge date. We remove duplicate admission records by subject and admission ID; missing clinical measurements alone do not exclude a patient. We order patients by subject, admission time, and admission ID; the last hospitalization is the index admission. Time zero is its `dischtime`, and death dates come from `patients.dod`. Invasive mechanical ventilation (IMV) is an inpatient exposure indicated by codes 9671, 9672, $5A1935Z$, $5A1945Z$, and $5A1955Z$. We define d_i as the recorded death date minus the discharge date, in days. The survival pair is $(T_i, \delta_i) = (d_i, 1)$ for a recorded death with $0 < d_i \le 365$; otherwise it is $(365, 0)$, using a 365-day administrative follow-up limit. We fit Cox to the observed event/censoring pairs and then use it to generate benchmark risk trajectories. The Cox covariates include current-admission demographic/administrative fields (age, sex, race, insurance, language, marital status, admission type, and admission source), prior-admission history (admission count, time since previous discharge, prior length of stay, diagnosis count, procedure count, and prior IMV), and previous-admission clinical fields (SpO2, heart rate, systolic/diastolic/mean blood pressure, three GCS fields, temperature, respiratory rate, creatinine, white-cell count, glucose, sodium, hemoglobin, and hematocrit). We summarize repeated valid measurements of a clinical field within the previous admission by their median, excluding measurements after that admission's discharge. We then impute missing numeric summaries with training-set medians and supply missingness indicators before standardization. For categorical fields, we use mode imputation and one-hot encoding; we merge categories with training frequency below 1% and unseen validation/test categories into `Other`. We fit all preprocessing on training patients. We retain 24 dimensions using training-set PCA, followed by standardization using training statistics. We fit a penalized Cox model on the training split only to the observed survival pairs (T_i, δ_i), with penalizer 0.1, then freeze it before target construction. We denote by $H_0(t)$ its interpolated baseline cumulative hazard and by $\eta_i^{(a)}$ the fitted Cox linear predictor for patient i with IMV covariate value $a \in \{0, 1\}$. Writing a_i^{obs} for the observed IMV status, we define

$$\begin{aligned} H_i^{\mathrm{obs}}(t) &= H_0(t)\exp\left(\eta_i^{(a_i^{\mathrm{obs}})}\right), \\ H_i^{(a)}(t) &= \begin{cases} H_i^{\mathrm{obs}}(t), & t \le t_0, \\ H_i^{\mathrm{obs}}(t_0) + \big(H_0(t) - H_0(t_0)\big)\exp\left(\eta_i^{(a)}\right), & t > t_0, \end{cases} \\ R_i^{(a)}(t) &= 1 - \exp\left(-H_i^{(a)}(t)\right). \end{aligned}$$

For the Cox fit, we use the prespecified nonlinear basis $h(u \mid z, a) = h_0(u)\exp\{\beta^\top \phi(z, a)\}$ with 109 components: the 24 PCA coordinates, squares and tanh transforms of the first 12 coordinates,

pairwise interactions among the first 8, a treatment main effect, treatment interactions with the first 16 linear coordinates, and treatment interactions with the first 8 square and tanh terms. The fitting interface is `lifelines.CoxPHFitter`. We evaluate the fitted model on an intermediate risk grid to construct the regime-specific target trajectory from the observed survival pairs. Using the frozen model, we construct two suffixes over $t = 51, \ldots, 200$ by imposing the fixed all-zero and all-one continuations. We linearly interpolate the baseline cumulative hazard on the daily grid; we use days 1–200, with the first 50 days as the prefix and days 51–200 as the scored suffix, with one day per step. The predictor interface receives the first three PCA coordinates, normalized time, lagged risk, treatment, and the corresponding treatment/risk history; the remaining 21 PCA coordinates affect the generated prefix only through the Cox risk and are not passed as extra predictor inputs. We open the post-t_0 target suffixes only after final training weights, configuration choices, and the TAM threshold are frozen, for final paired scoring. This benchmark uses a fitted Cox target generator for semi-synthetic model recovery.

Implemented coordinate behavior. For synthetic data, $X_{1:3}$ are carried, $X_{4:5}$ are predicted and may be corrected from the last reconstructed row, X_6 is hidden and never provided to TSA, A follows the fixed continuation, and Y is predicted and scored but is not written back by TAM. In the MIMIC-derived path, $x_{1:3}$ are carried PCA features and $x_4 = t/200$ is assembled deterministically at every index; it is neither predicted by TSA nor written back by TAM. The lag-risk coordinate is constructed by ordinary autoregression: $x_{5,t+1} = \widehat{R}_t$. We assemble time and lag-risk for the new row; existing corrected rows remain in the window. At each step we append the reported risk and generated state to the rolling window. If TAM triggers, its masked reconstruction updates the allowed generated x_5 entry in that window; the resulting window is then used to predict the next step. Time advances deterministically. We protect static features, the prescribed treatment, the observed prefix, and the already reported $Y = \widehat{R}_t$. The feedback operation is the window update in Equation (3).

Common Evaluation Contract. For each continuation, we compute performance over its complete 150-step suffix using NMAE and NRMSE. For each seed, we first aggregate subject- and time-level errors over the suffix, then report the mean $\pm$ sample standard deviation across 30 paired seeds, with lower values indicating better predictive accuracy. We fix subject-level splits and compute normalization statistics from training subjects only. We select context lengths of $l = 10$ for synthetic scenarios and $l = 20$ for the MIMIC benchmark. We freeze model-specific configurations and inference-time thresholds before post-t_0 targets are opened for test scoring.

E Algorithms and Module Schematics

In the algorithms, we specify TSA/TAM training and recursive application for the two action-specific predictor parameter sets θ_0, θ_1. A supplied binary continuation $\bar{a} = (a_{t_0+1}, \ldots, a_T)$ may be constant or time-varying; $g_\theta(W, a) := g_{\theta_a}(W)$, and the branches have no shared updates. In the experiments, we make separate all-zero and all-one calls.

We summarize the two modules in Figure 6 and detail their layer specifications and training procedures below.

For each $a \in \{0, 1\}$, $\mathcal{T}_{\mathrm{tr}}^{(a)}$ uses source-only prefix windows $W_{i,t-1}^{l}$ from $t - l, \ldots, t - 1$ to target t, with $l + 1 \le t \le 50$, after filtering $A_{i,t} = a$:

$$\mathcal{T}_{\mathrm{tr}}^{(a)} = \{(i, t) : i \in I_{\mathrm{tr}},\ l + 1 \le t \le 50,\ A_{i,t} = a\}\,.$$

Before action filtering, each patient contributes $50 - l$ candidate pairs (40 for $l = 10$, 30 for $l = 20$). For validation, we use the same rule on I_{val}; branch updates remain separate and post-t_0 records enter neither TSA nor TAM fitting. We let $\mathbf{u}_{\theta_a}(W, A) \in \mathbb{R}^{64}$ denote its representation after the temporal stack, and let $\widehat{S}_{i,t}^{(a)} = h_{\theta_a,S}(\mathbf{u}_{\theta_a}(W_{i,t-1}^{l}, A_{i,t})) \in \mathbb{R}^2$ be the state-head output, and let $S_{i,t} \in \mathbb{R}^2$ denote its true next-state target. The outcome head receives the true next state during training, so

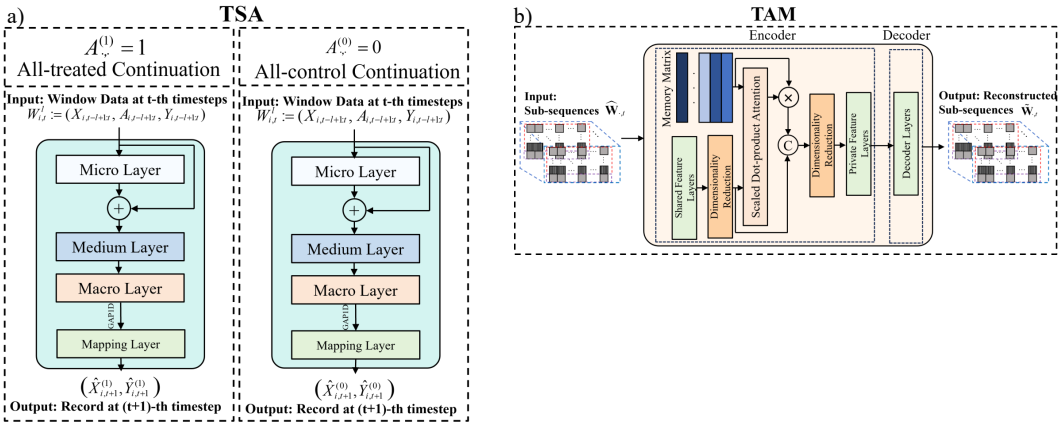


Figure 6: **TSA and TAM module architectures.** (a) Each action-specific TSA branch processes the input window through residual convolution (micro), an LSTM (medium), self-attention (macro), and state/outcome mappings. (b) TAM reconstructs a block of windows through shared encoding, prototype retrieval, fusion, a private adapter, and decoding. During online adaptation, we update only the private adapter and keep the monitor frozen. Selected reconstructed entries update the operational window according to Equation 3.

$\widehat{Y}^{\mathrm{TF},(a)}_{i,t} = h_{\theta_a,Y}([\mathbf{u}_{\theta_a}; S_{i,t}; A_{i,t}])$ has 67 inputs. The branch losses are

$$\mathcal{L}_S^{(a)}(\theta_a) = \frac{1}{|\mathcal{T}_{\mathrm{tr}}^{(a)}|} \sum_{(i,t)\in\mathcal{T}_{\mathrm{tr}}^{(a)}} \frac{1}{2}\|\widehat{S}^{(a)}_{i,t} - S_{i,t}\|_2^2, \tag{4}$$

$$\mathcal{L}_Y^{\mathrm{TF},(a)}(\theta_a) = \frac{1}{|\mathcal{T}_{\mathrm{tr}}^{(a)}|} \sum_{(i,t)\in\mathcal{T}_{\mathrm{tr}}^{(a)}} (\widehat{Y}^{\mathrm{TF},(a)}_{i,t} - Y_{i,t})^2, \tag{5}$$

$$\mathcal{L}_{\mathrm{TSA}}^{(a)} = \mathcal{L}_S^{(a)} + \mathcal{L}_Y^{\mathrm{TF},(a)} + 10^{-5}\Omega_{\mathrm{TSA}}(\theta_a). \tag{6}$$

We give the state and teacher-forced outcome terms equal unit weight. At inference, synthetic outcome heads receive predicted states; on MIMIC, the dispatch assembles the deterministic time and lag-risk slots described in Appendix D, then combines carried coordinates with branch-specific outputs.

For the residual CNN, we use two same-padded one-dimensional convolutions with kernel width 3 and 64 channels, plus a kernel-width-1 residual projection. We apply LayerNorm and GELU after the first convolution and add the second convolution to the residual branch. The medium layer is one 64-unit LSTM retaining its full sequence output. For the macro layer, we use one four-head self-attention module with key dimension 16 per head, followed by LayerNorm and global average pooling. A shared $64 \to 64$ GELU map feeds a linear $64 \to 2$ state head. In the outcome head, we apply $67 \to 64$ GELU and a linear output. We use Adam without a learning-rate schedule, AdamW-style weight decay, or gradient clipping. We apply kernel-L_2 to the two main convolutions and the shared/outcome mappings with coefficient 10^{-5}, and use dropout 0.1 only in the attention and outcome mappings. We retain final branch weights, with supported branch-local choices selected on each branch's validation data.

After the final TSA branch weights are saved, we train the complete TAM offline. We build training blocks $\mathcal{B}_{\mathrm{tr}}$ from each training patient's observed records and treatments in the first 50 steps, and build validation blocks $\mathcal{B}_{\mathrm{val}}$ analogously from each validation patient's prefix. Within each patient, we group unit-stride overlapping complete length-l windows into blocks of $N' = 10$ successive windows; these blocks use no generated TSA suffix. We estimate normalization statistics from training prefixes only. We use validation reconstruction loss for early stopping and restored-weight selection, and validation full-block reconstruction discrepancies to calibrate δ_{corr}. Each block has shape $N' \times l \times d$, where $d = \dim(X) + 2$ and $D_{\mathrm{in}} = N'ld$. The shared modules are the encoder ϕ_{enc}, prototype memory $\mathbf{M}$, and shared decoder ϕ_{dec}. For a block $\mathbf{W}$, the encoder has three 64-unit GELU layers, the query is a linear projection followed by LayerNorm, and each raw 64-dimensional memory

Algorithm 1 Action-conditioned TSA training

Require: supplied branch training transitions $\mathcal{T}_{\mathrm{tr}}^{(a)}$ built from inputs $t-l,\dots,t-1$ and target t for $l+1 \le t \le 50$ with $A_{i,t}=a$, and validation transitions built by the same rule for $a \in \{0,1\}$

for each $a \in \{0,1\}$ **do**
  maintain branch parameters θ_a and initialize the two branch loss accumulators
  for each $(W_{i,t-1}^{l}, A_{i,t}, S_{i,t}, Y_{i,t}) \in \mathcal{T}_{\mathrm{tr}}^{(a)}$ **do**
    $\mathbf{u}_{i,t} \leftarrow \mathbf{u}_{\theta_a}(W_{i,t-1}^{l}, A_{i,t})$
    $\widehat{S}_{i,t}^{(a)} \leftarrow h_{\theta_a,S}(\mathbf{u}_{i,t})$
    $\widehat{Y}_{i,t}^{\mathrm{TF},(a)} \leftarrow h_{\theta_a,Y}([\mathbf{u}_{i,t}; S_{i,t}; A_{i,t}])$
    accumulate $\frac{1}{2}\|\widehat{S}_{i,t}^{(a)} - S_{i,t}\|_2^2$ in $\mathcal{L}_S^{(a)}$
    accumulate $(\widehat{Y}_{i,t}^{\mathrm{TF},(a)} - Y_{i,t})^2$ in $\mathcal{L}_Y^{\mathrm{TF},(a)}$
  end for
  optimize only θ_a with $\mathcal{L}_S^{(a)} + \mathcal{L}_Y^{\mathrm{TF},(a)} + 10^{-5}\Omega_{\mathrm{TSA}}(\theta_a)$
  save the final branch weights $\theta_{a,\mathrm{final}}$
  select supported branch-local configuration choices using its validation data only
end for
dispatch $g_\theta(W,a) = g_{\theta_{a,\mathrm{final}}}(W)$ before recursive evaluation

row is subject to the offline unit-norm constraint before the memory-side LayerNorm. Prototype retrieval is denoted by $\mathbf{r} = \mathcal{R}(\mathbf{q}, \mathbf{M})$; concatenating query and retrieval gives a $128 \to 64$ GELU fusion map. A private three-layer $64 \to 64$ GELU adapter feeds a decoder with two 64-unit GELU layers and a linear output restoring the original D_{in} block shape. These modules form a single serial reconstruction path. The explicit path is

$$\mathbf{z}_{\mathrm{enc}} = \phi_{\mathrm{enc}}(\mathrm{flat}(\mathbf{W})), \qquad \mathbf{q} = \mathrm{LN}(W_q \mathbf{z}_{\mathrm{enc}} + b_q), \qquad \mathbf{h} = \mathrm{GELU}(W_f[\mathbf{q}; \mathcal{R}(\mathbf{q},\mathbf{M})] + b_f),$$

where, for K memory rows, $\bar{\mathbf{m}}_k = \mathrm{LN}_m(\mathbf{M}_k)$ and $\alpha_k = \mathrm{softmax}_{k=1:K}(\mathbf{q}^\top \bar{\mathbf{m}}_k / \sqrt{64})$, $\mathcal{R}(\mathbf{q},\mathbf{M}) = \sum_{k=1}^{K} \alpha_k \bar{\mathbf{m}}_k$. The serial reconstruction is

$$\widetilde{\mathbf{W}}(\phi_{\mathrm{priv}}) = \phi_{\mathrm{dec}}(\phi_{\mathrm{priv}}(\mathbf{h})).$$

Algorithm 2 Offline serial TAM training and validation calibration

Require: final TSA branch weights, training blocks $\mathcal{B}_{\mathrm{tr}}$, validation blocks $\mathcal{B}_{\mathrm{val}}$, configured minibatch and epoch budgets, and patience

initialize $\phi_{\mathrm{enc}}, \mathbf{M}, \phi_{\mathrm{priv}}, \phi_{\mathrm{dec}}$
for each configured offline epoch **do**
  for each minibatch $\mathcal{Q} \subset \mathcal{B}_{\mathrm{tr}}$ **do**
    compute $\widetilde{\mathbf{W}} \leftarrow \mathrm{Rec}_{\mathrm{TAM}}(\mathbf{W})$ for each $\mathbf{W} \in \mathcal{Q}$, where $\mathbf{W} \in \mathbb{R}^{N' \times l \times d}$
    compute the minibatch mean $\mathcal{L}_{\mathrm{off}}(\mathcal{Q}) \leftarrow |\mathcal{Q}|^{-1} \sum_{\mathbf{W} \in \mathcal{Q}} D_{\mathrm{in}}^{-1} \| \mathrm{flat}(\widetilde{\mathbf{W}}) - \mathrm{flat}(\mathbf{W})\|_2^2 + 10^{-5}\Omega_{\mathrm{ker}}$
    update all TAM parameters with Adam using $\mathcal{L}_{\mathrm{off}}(\mathcal{Q})$ and enforce the unit-norm prototype constraint
  end for
  evaluate validation val_loss and apply the configured patience rule
end for
restore the complete offline TAM, including $\phi_{\mathrm{priv}}^{\mathrm{off}}$, with the lowest validation val_loss
set $\delta_{\mathrm{corr}} = \mu_{\mathrm{val}} + 3s_{\mathrm{val}}$ from validation full-block mean-squared reconstruction discrepancies
freeze encoder, prototypes, fusion map, decoder, $\phi_{\mathrm{priv}}^{\mathrm{off}}$, threshold, and fixed $N' = 10$ before recursive evaluation

For each eligible block, the frozen offline TAM computes the full-block gate score. If it fires, we use a newly constructed Adam optimizer to update the independent private-adapter copy for five steps using reconstruction MSE normalized by D_{in}; the frozen encoder, prototypes, fusion map, decoder, monitor adapter, and TSA remain unchanged. For each patient–continuation call, we initialize the adaptation state from offline weights and retain it across triggers in that call.

Algorithm 3 Recursive prediction for one patient under a specified intervention continuation

Require: observed W^l_{i,t_0}, prescribed $\bar{a} = (a_{t_0+1}, \ldots, a_T)$, final branch weights $\theta_{a,\text{final}}$, frozen complete offline TAM monitor, δ_{corr}, fixed $N' = 10$, and writeback configuration

$\widehat{W}^l_{i,t_0} \leftarrow W^l_{i,t_0}$
create an adaptation private state $\phi^{i,\bar{a}}_{\text{adapt}}$ from the restored offline private weights
initialize the generated-window buffer $\mathcal{B} \leftarrow \varnothing$
for $t = t_0 + 1, \ldots, T$ **do**
  for MIMIC, construct new-row time $x_{4,t} = t/200$ and lag-risk from the preceding reported risk; preserve existing rows in the current corrected window $\widehat{W}^l_{i,t-1}$ used by TSA
  $\widehat{Z}_{i,t} \leftarrow g_{\theta_{\text{final}}}(\widehat{W}^l_{i,t-1}, a_t)$, where $\theta_{\text{final}} = (\theta_{0,\text{final}}, \theta_{1,\text{final}})$
  record $\widehat{Z}_{i,t}$ as the reported emission before Shift or any TAM writeback
  $\overline{W}^l_{i,t} \leftarrow \text{Shift}(\widehat{W}^l_{i,t-1}, (\widehat{Z}_{i,t}, a_t))$
  append the resulting complete window to $\mathcal{B}$
  if $|\mathcal{B}| < N'$ **then**
    $\widehat{W}^l_{i,t} \leftarrow \overline{W}^l_{i,t}$
  else
    $\mathbf{W}_{i,t} \leftarrow \mathcal{B} \in \mathbb{R}^{N' \times l \times d}$
    $\widetilde{\mathbf{W}}^{\text{gate}}_{i,t} \leftarrow \text{FrozenTAM}_{\text{off}}(\mathbf{W}_{i,t})$
    $d_{i,t} \leftarrow D_{\text{in}}^{-1} \| \text{flat}(\widetilde{\mathbf{W}}^{\text{gate}}_{i,t}) - \text{flat}(\mathbf{W}_{i,t}) \|_2^2$
    if $d_{i,t} > \delta_{\text{corr}}$ **then**
      keep the complete offline TAM monitor and $g_{\theta_{\text{final}}}$ frozen
      hold $\mathbf{W}_{i,t}$ fixed as the adaptation target
      for $k = 1, \ldots, 5$ **do**
        $\mathcal{L}_{\text{self}} \leftarrow D_{\text{in}}^{-1} \| \text{flat}(\widetilde{\mathbf{W}}^{\text{prop}}_{i,t}(\phi^{i,\bar{a}}_{\text{adapt}})) - \text{flat}(\mathbf{W}_{i,t}) \|_2^2$
        take one step of the private-adapter optimizer on $\mathcal{L}_{\text{self}}$, updating only $\phi^{i,\bar{a}}_{\text{adapt}}$
      end for
      recompute $\widetilde{\mathbf{W}}^{\text{prop}}_{i,t}$ with the adaptation private state; the frozen monitor is not changed
      let $\widetilde{W}^{\text{last}}_{i,t} \leftarrow (\widetilde{\mathbf{W}}^{\text{prop}}_{i,t})_{N'}$ and let $\widetilde{R}^{\text{last}}_{i,t}$ be its last generated row
      $P_{i,t} \leftarrow \text{Lift}_l(\widetilde{R}^{\text{last}}_{i,t}, \overline{W}^l_{i,t})$
      form $M_{i,t}$ with zeros outside that last row and with static, treatment, observed-prefix, and reported-outcome entries protected; select synthetic $X_4{:}X_5$ or MIMIC x_5 only
      $\widehat{W}^l_{i,t} \leftarrow \overline{W}^l_{i,t} + M_{i,t} \odot (P_{i,t} - \overline{W}^l_{i,t})$
    else
      $\widehat{W}^l_{i,t} \leftarrow \overline{W}^l_{i,t}$
    end if
    clear the window buffer
  end if
end for
return $\widehat{\mathbf{Z}}^{\bar{a}}_{i,t_0+1:T} \leftarrow (\widehat{Z}_{i,t_0+1}, \ldots, \widehat{Z}_{i,T})$

We clear the buffer after each complete N'-append block; its length-l windows may overlap. We return the recorded $\widehat{Z}_{i,t}$ sequence, and writeback affects only later TSA inputs while actions stay fixed. Frozen(0) uses the same gate and mask without private updates; TAM-off/TSA-only skips both gate and writeback.

The experimental calls are separate:

$$\bar{a}^{(0)} = (0, \ldots, 0), \qquad \widehat{\mathbf{Z}}^{\bar{a}^{(0)}}_{i,t_0+1:T} \leftarrow \text{RecursivePredict}(W^l_{i,t_0}, \bar{a}^{(0)}),$$

$$\bar{a}^{(1)} = (1, \ldots, 1), \qquad \widehat{\mathbf{Z}}^{\bar{a}^{(1)}}_{i,t_0+1:T} \leftarrow \text{RecursivePredict}(W^l_{i,t_0}, \bar{a}^{(1)}).$$

F BASELINES

The benchmark specification lists seven displayed baselines from three families. MSM and SNMM are longitudinal estimators; RMSN, CRN, G-Net, and G-Transformer are neural sequence and g-computation models; and GALILEO is a learned-dynamics model. For the main baseline comparison, we evaluate all methods under the all-zero and all-one continuations. Table 5 records the displayed baseline interfaces.

Table 5: Specification of the seven displayed baselines used in the common trajectory benchmark.

Method	**Family**	**Benchmark interface**
MSM (Robins et al., 2000)	Longitudinal estimation	Stabilized sequential-IPW marginal outcome regression plus a fitted history-based covariate transition for recursive fixed-regime outcome prediction.
SNMM (Robins, 1994)	Longitudinal estimation	Propensity-adjusted g-estimated structural blip plus a history-based covariate transition for recursive fixed-regime outcome prediction.
RMSN (Lim, 2018)	Neural sequence	Propensity-weighted recurrent encoder-decoder; the prescribed continuation is supplied during decoding and recursive prediction, which returns an outcome suffix.
CRN (Bica et al., 2020b)	Neural sequence	Balanced history representation with an action-conditioned decoder; recursively evaluates the prescribed continuation.
G-Net (Li et al., 2020)	Neural sequence/g-computation	Recurrent g-computation with ordered conditional state and outcome factors; recursively feeds the generated state under the prescribed continuation.
G-Transformer (Xiong et al., 2024)	Neural sequence/g-computation	Transformer-based history/transition model with the prescribed continuation and recursive outcome-suffix generation.
GALILEO (Chen et al., 2023)	Learned dynamics	Learned-dynamics counterfactual environment model evaluated under an externally fixed continuation rather than by policy optimization; it returns an outcome suffix.

Baseline implementation and optimization. For the displayed comparison, we use benchmark implementations under the common evaluation interface. Following Robins et al. (2000), we implement MSM locally in NumPy as stabilized sequential-IPW weighted regression with a recursive covariate transition model, using Ridge 0.01 and a regularized logistic propensity model with at most 250 Newton updates and maximum absolute step below 10^{-9}; it is full-batch with no epoch-style early stopping or validation checkpoint. Following Robins (1994), we implement SNMM through g-estimation in NumPy with auxiliary and recursive covariate regressions, the same Ridge 0.01 and propensity solver, and no validation checkpoint selection.

We implement RMSN following Lim (2018) with benchmark-specific GRU propensity networks, encoder, and decoder of width 32; training uses Adam with learning rate 0.002, batch size 128, at most 300 epochs per training stage, and gradient clipping at 5. We implement CRN following Bica et al. (2020b) with a single-layer width-32 LSTM encoder and decoder with a balanced representation, gradient-reversal treatment head, and state/outcome heads; it uses Adam 0.002, batch size 128, at most 300 epochs, and gradient clipping at 5. We implement G-Net following Li et al. (2020), using a single-layer width-32 LSTM with ELU mapping, predict the dynamic state before the outcome, and use the same Adam 0.002, batch-128, 300-epoch maximum, and clipping at 5.

For G-Transformer, we adapt the authors' implementation (Xiong et al., 2024), using MSE training and empirical-residual sampling. Its two-layer Transformer has width 64, four heads, FFN width 128, and dropout 0.1. We use Adam with learning rate 10^{-3}, batch size 128, weight-L_2 10^{-5}, and gradient clipping at 5; training runs for at most 300 epochs. We obtain predictions by averaging 100 sampled paths on the outcome scale.

For GALILEO, we adapt the official implementation (Chen et al., 2023) to binary treatments, retaining TRPO, weighted likelihood, and two discriminators. We train for at most 300 outer iterations. At test time, the prescribed continuation supplies treatment.

For RMSN, CRN, G-Net, G-Transformer, and GALILEO, we use validation NMAE for checkpoint selection, stop after 20 consecutive validation checks without improvement, and restore the checkpoint

with the lowest validation NMAE. Training budgets are counted in epochs (per stage for RMSN), except for GALILEO, which uses outer iterations.

For all seven baselines, we use the same subject split, binary-treatment interface, observed prefix through $t_0 = 50$, fixed all-zero/all-one recursive evaluation, 150-step suffix, and scoring rule. On MIMIC, we limit inputs to the first three PCA coordinates, normalized time, lagged risk, treatment, and treatment/risk history; Cox coefficients, the other 21 PCA coordinates, and future targets are excluded. Each method uses the stated observable inputs and its listed training, configuration-selection, and checkpoint rules for final scoring.

For each subject, we generate all-zero and all-one trajectories separately over $H = 150$ post-decision steps. We score the predicted outcome trajectory; validation selects models/configurations, and test targets open only for final scoring. We compute NMAE and NRMSE per regime. Within each seed, we aggregate trajectory errors over subjects and time, then report them as mean $\pm$ sample SD across 30 paired seeds.

G Window Length

The context-length study follows the validation-only selection procedure used to determine the main experimental configurations. For the compact nonlinear synthetic benchmark, we evaluate $l \in \{5, 10, 20\}$; for the MIMIC-IV-derived benchmark, the same base grid is supplemented with the prespecified boundary diagnostic $l = 30$, giving $l \in \{5, 10, 20, 30\}$. For each context, we use the same dataset-specific partitions and five paired seeds $\{0, 6, 12, 18, 24\}$. We minimize the mean validation NMAE across these five seeds for context selection, breaking ties in favor of the shorter window. For selection, we use one-step factual-prefix targets only, with no post-t_0 or test outcomes opened. Using the locked procedure, we select $l = 10$ for the synthetic benchmark and $l = 20$ for the MIMIC-IV-derived benchmark; we freeze these values before the all-zero and all-one test trajectories are scored.

In the two panels, we report the synthetic grid (Table 6) and the MIMIC-IV-derived grid (Table 6); in the latter, $l = 20$ has the lowest mean NMAE and NRMSE, whereas $l = 30$ is higher on both.

Table 6: Context-length selection under validation-only evaluation. Panel (a) is the compact Synthetic (nonlinear outcome) benchmark; panel (b) is MIMIC-derived Cox recovery. We report validation NMAE and NRMSE as mean $\pm$ sample standard deviation across five paired seeds.

(a) Synthetic (nonlinear outcome)

Metric	$l = 5$	$\mathbf{l = 10}$	$l = 20$
NMAE	0.229 ± 0.018	0.224 ± 0.015	0.227 ± 0.017
NRMSE	0.288 ± 0.022	0.282 ± 0.019	0.285 ± 0.021

(b) MIMIC-derived Cox recovery

Metric	$l = 5$	$l = 10$	$\mathbf{l = 20}$	$l = 30$
NMAE	0.318 ± 0.047	0.296 ± 0.052	0.274 ± 0.041	0.337 ± 0.063
NRMSE	0.436 ± 0.061	0.412 ± 0.068	0.381 ± 0.055	0.489 ± 0.079

H Hardware-Specific Computational Profile

Measurement Protocol. We profile LTTS on one exclusive NVIDIA GeForce RTX 4090 D using the seed-0 checkpoint, its locked context length, a fixed representative input shape, and an $H = 150$ recursive prediction. In each of three fresh processes, we execute 20 warm-ups per operation and then measure 100 CUDA-synchronized repetitions. We summarize the pooled 300 repetitions by the mean and sample standard deviation, median, and 95th percentile. We record peak host resident memory for the same operations. For each repetition, we restore the same checkpoint and input state; we reset private-adapter state and memory counters before measurement. Training-batch timings include the forward pass, backward pass, and optimizer update in the audited timed training step. In Table 7, we record one TSA-branch, TAM, private-adapter, and Full LTTS recursive-prediction operations; the recursive row measures the complete LTTS execution from the same input and checkpoint state.

The recursive row is the full $H = 150$ rollout, including TSA prediction, rolling complete-block monitoring, trigger-time private adaptation, and feedback writeback; timings are specific to the audited hardware and protocol. For the lower-panel timings, we use the same device and input-evaluation procedure across the three benchmarks and report mean elapsed time in seconds. The upper panel profiles isolated operations, whereas the lower panel measures end-to-end rollout time; these are separate measurements rather than components of a timing decomposition. We measure Full(5)

Table 7: Hardware-specific computational profile of LTTS on one NVIDIA GeForce RTX 4090 D. We compute timing statistics from 300 CUDA-synchronized repetitions after excluding warm-up executions. The host RSS entry is the maximum over the same pooled repetitions. The recursive-prediction row is the complete Full LTTS execution timing. In the lower panel, we report matched per-patient $H = 150$ rollout means in seconds and pooled trigger rates over actual monitoring opportunities for the two constant arms.

Stage	Operation or configuration	Mean ± SD (ms)	Median (ms)	p95 (ms)	Host RSS (MiB)
Training	One TSA-branch forward/backward/optimizer batch	131.71 ± 39.50	113.86	188.83	1,761.9
Training	TAM module training batch	75.09 ± 14.93	70.00	114.18	1,764.3
Inference	One TSA-branch prediction step	36.65 ± 9.28	42.60	47.63	1,764.6
Inference	TAM 10-step monitoring segment	10.98 ± 3.16	8.63	15.98	1,764.7
Inference	Private-adapter update during recursive prediction	277.48 ± 74.64	324.40	373.62	1,917.4
Recursive prediction	Full LTTS recursive prediction, $H = 150$	$29{,}578.48 \pm 2{,}741.26$	29,796.16	33,652.71	2,523.6

Per-patient 150-step rollout and pooled trigger rates

Benchmark	TSA-only (s)	Frozen(0) (s)	Full(5) (s)	Frozen trigger rate	Full trigger rate
Synthetic (linear outcome)	5.84	10.92	29.31	18.7%	14.3%
Synthetic (nonlinear outcome)	5.97	11.46	29.86	22.0%	17.3%
MIMIC-derived Cox recovery	6.18	12.07	30.24	24.7%	19.0%

rollout time at about 2.5–$2.7\times$ Frozen(0) and 4.9–$5.0\times$ TSA-only in these timings, quantifying the additional computation used for adaptive feedback.

I Component Ablations and Post-Trigger Diagnostics

For the component and backbone studies, we use the common full-scale benchmark, continuation, normalization, and validation-selection protocol; dataset construction and seed-level aggregation are specified in Appendices D and J.3.

I.1 Full-Scale Cross-Benchmark Component Ablation

In Table 8, we report the matched continuation-specific component comparison.

Table 8: Full-scale component ablations across Synthetic (linear outcome), Synthetic (nonlinear outcome), and MIMIC-derived Cox recovery. In all rows, we report outcome-trajectory error as mean ± sample SD over 30 paired seeds, with each value a per-seed equal-weight $1/3$ macro-average of the independently normalized dataset metrics. The Full LTTS means also agree, up to rounding, with the equal-weight average of the dataset means in Table 2.

Condition	All-zero NMAE	All-zero NRMSE	All-one NMAE	All-one NRMSE
Full LTTS	0.295 ± 0.030	0.399 ± 0.053	0.255 ± 0.031	0.354 ± 0.045
w/o TAM	0.655 ± 0.084	0.748 ± 0.096	0.609 ± 0.099	0.701 ± 0.112
w/o Mi-L	0.333 ± 0.020	0.417 ± 0.027	0.569 ± 0.219	0.687 ± 0.243
w/o Me-L	0.341 ± 0.012	0.426 ± 0.015	0.405 ± 0.105	0.501 ± 0.119
w/o Ma-L	0.333 ± 0.018	0.416 ± 0.031	0.389 ± 0.066	0.480 ± 0.077
TSA + fixed clipping	0.381 ± 0.052	0.472 ± 0.071	0.394 ± 0.061	0.501 ± 0.083
TSA + prototype L_2	0.417 ± 0.063	0.508 ± 0.078	0.436 ± 0.072	0.552 ± 0.091

In the TSA ablation, we remove the micro-, medium-, and macro-connectivity layers one at a time. The labels Mi-L, Me-L, and Ma-L refer to the residual CNN, LSTM, and self-attention modules, respectively. For each variant, we preserve the output head, action input, loss, optimizer, standardization, context length, TAM interface, and recursive prediction protocol. We refit the three layer-removal variants independently for all 30 seeds under this matched protocol. We replace TAM with two controls while retaining TSA and the prescribed interventions. For clipping, we bound each permitted generated feedback coordinate by $[\mu_j - k\sigma_j, \mu_j + k\sigma_j]$, using training means and standard deviations. The grid is $k \in \{2, 3, 4, 5, \infty\}$; we use validation-prefix recursive NMAE to select one value per dataset and seed, shared by both actions, with ties favoring larger k.

For the prototype-L_2 control, we use a fixed target in standardized feedback coordinates. We concatenate the mean query over training blocks, q_{ref}, with each LayerNorm-transformed prototype and pass it through the frozen fusion map, offline private adapter, and decoder. From each reconstructed block,

we extract the permitted coordinates of the last row of the last window; their equal-weight mean over prototypes is $\bar{p}$. Feedback is $z^\star = (z_{\text{pred}} + \lambda\bar{p})/(1+\lambda)$, the minimizer of $\|z - z_{\text{pred}}\|_2^2 + \lambda\|z - \bar{p}\|_2^2$. For the grid $\lambda \in \{0, 0.01, 0.1, 1, 10\}$, we use the same validation criterion and selection granularity as clipping, with ties favoring smaller λ. We keep all modules frozen; this fixed target differs from TAM's query-dependent retrieval. Both controls protect treatment and recorded outputs. Their control parameters are distinct from TAM's offline kernel-L_2 coefficient of 10^{-5}.

I.2 Frozen, Local-Correction, and Action-Switch Evidence

Predicted-state diagnostic and calibrated frozen control. To diagnose the training-input mismatch, we replace the true next state by a stop-gradient prediction only at the outcome head in the synthetic predicted-state (PS) diagnostic. The history windows are factual training-prefix windows from steps 1:50. We keep state/outcome targets, equal-weight MSE, action filtering, and the structural budget unchanged. We train the primary TSA with true next states and evaluate PS as a diagnostic variant.

The mixing coefficient α controls writeback strength; $\alpha = 0$ leaves the operational input unchanged. For each dataset, training variant, and seed, we jointly select (α_0, α_1) for calibrated Frozen from $\{0, .25, .5, .75, 1\}^2$ (25 pairs) on validation prefixes. For synthetic validation, we use true states at 1:10 then recursion at 11:50; for MIMIC, we use true states at 1:20 then recursion at 21:50. Rollouts follow recorded factual actions; true dynamic X/Y after recursion starts are scoring-only. For each candidate, we reset the patient session and keep the same frozen gate/block schedule/mask. The objective is equal-weight two-arm NMAE with training normalization; ties within 10^{-8} choose smaller $\alpha_0 + \alpha_1$, then smaller α_0. At an eligible trigger (the forced local correction is described separately below), writeback after step t uses

$$W_{i,t}^{l,+} = \overline{W}_{i,t}^{l} + \alpha_{A_{i,t+1}} M_{i,t} \odot \big(\widetilde{W}_{i,t}^{\text{Frozen}} - \overline{W}_{i,t}^{l}\big).$$

We fix the validation-selected coefficients for both the complete 150-step and local 20-step evaluations. Full(5) retains five private updates and has no additional α. We use Frozen(α^*) for this control: the 0 in Frozen(0) denotes zero private-update steps, α_0 is the action-0 coefficient.

In these panels, we distinguish frozen writeback from private adaptation under the primary fixed-arm checkpoints and the two specified nonlinear switches. Frozen(0) writes back the frozen-monitor proposal with zero private updates; Full(5) uses five, so endpoint gates may differ. TAM-off/TSA-only skips both.

For the forced single-correction diagnostic in Table 9b and Table 3b, we use the original TSA and retain 120 uniformly sampled valid positions per seed and setting (3,600 per setting; 14,400 total), with candidate depth $h \le 130$. We use TSA-only to generate the preceding trajectory and window blocks. At each sampled position, we copy the common state and block to all branches. For the Full branch, we initialize the private adapter from the offline weights and perform five updates with a fresh Adam optimizer. We compare the three correction branches with no correction; all preserve treatment/protected coordinates and use TAM-off for the next 20 steps. With this diagnostic, we measure the subsequent effect of one correction. We average NMAE per seed; window percentages are strict operational-error reductions, separate from future-20 NMAE and natural trigger rates.

In panel (c), we use the nonlinear generator with the specified $0 \to 1$ or $1 \to 0$ switch at $t = 101$, rolling generated history, and actual sequential-DGP reference; no reset or reference input occurs at the switch. The post-switch column covers $h = 51{:}70$.

I.3 Internal TAM Controls

In Figure 3, we examine trigger policy, prototype retrieval, and adaptation budget under the same fixed TSA checkpoints, subject splits, prefixes, all-zero/all-one continuation arms, $H = 150$, training-split normalization, and 30 paired seeds. TSA-only bypasses both private adaptation and TAM writeback, so normal autoregressive state feedback continues. For random-trigger TAM (5), we replace the error-based gate with a validation-calibrated randomized gate matched to the expected trigger budget; we retain the masks and five private steps and record realized trigger counts. The comparison matches the validation-based expected budget; realized test counts depend on each resulting trajectory. For TAM without prototype (5), we replace the retrieval vector with zero, keep the fusion and remaining

Table 9: **Additional evidence for writeback and adaptation.** In panel (a), we report complete 150-step NMAE for Frozen(0) and calibrated Frozen(α^*); Frozen(0) has zero private updates and Full(5) has five. In panel (b), we report future-20 NMAE for no correction and Frozen(0), plus strict window-error reduction percentages for Frozen(0)/Full. Calibrated Frozen and Full future-20 NMAE are in Table 3b. In panel (c), we report total and first-20 post-switch NMAE. Entries are means $\pm$ sample SD over 30 paired seeds.

(a) Fixed-arm Frozen controls

Benchmark–arm	Frozen(0) NMAE	Frozen(α^*) NMAE
Linear, $a = 0$	0.681 ± 0.092	0.621 ± 0.084
Linear, $a = 1$	0.596 ± 0.071	0.572 ± 0.071
Nonlinear, $a = 0$	0.718 ± 0.107	0.604 ± 0.079
Nonlinear, $a = 1$	0.652 ± 0.086	0.551 ± 0.068
MIMIC-derived, $a = 0$	0.624 ± 0.083	0.593 ± 0.086
MIMIC-derived, $a = 1$	0.572 ± 0.096	0.542 ± 0.077

Full(5) values are in Table 2; the calibrated column uses the validation-only coefficient rule above.

(b) Forced single-correction diagnostic (synthetic)

Benchmark–arm	No NMAE	Frozen(0) NMAE	Frozen(0) window ↓	Full window ↓
Linear, $a = 0$	0.754 ± 0.103	0.681 ± 0.091	64.0%	72.7%
Linear, $a = 1$	0.667 ± 0.089	0.598 ± 0.076	61.3%	70.0%
Nonlinear, $a = 0$	0.803 ± 0.116	0.719 ± 0.101	60.0%	69.3%
Nonlinear, $a = 1$	0.741 ± 0.104	0.654 ± 0.090	58.7%	67.3%

(c) Nonlinear action-switch stress test

Switch	Total			Post 20		
	TSA	Frozen	Full	TSA	Frozen	Full
$0 \to 1$	0.985 ± 0.146	0.632 ± 0.093	0.356 ± 0.051	1.082 ± 0.174	0.714 ± 0.107	0.417 ± 0.064
$1 \to 0$	0.904 ± 0.132	0.611 ± 0.088	0.371 ± 0.056	1.011 ± 0.158	0.689 ± 0.102	0.435 ± 0.069

network, refit TAM offline, and calibrate the threshold with the same validation rule. For Full TAM (10), we change only per-trigger private updates from five to ten, so later triggers may change. For all TAM variants, we preserve the variable-specific feedback mask; error-triggered variants use the frozen offline monitor.

Figure 3 summarizes the six benchmark– continuation panels as means $\pm$ sample SD over 30 paired seeds.

I.4 Prediction Errors after the First Trigger

For the first-trigger closed-loop audit, we consider test patients from all three datasets, both fixed actions, and seeds 0–9. Along the Full LTTS path, we select the first legal trigger at which all $h \in \{1, 5, 10, 20\}$ are scoreable. Immediately before its writeback, we copy the same state into Full LTTS and TAM-off branches, matching patient, arm, seed, target, and loss. We use the same selected trigger point for both branches and determine eligibility by trigger validity and follow-up availability before outcomes are compared. Full LTTS continues subsequent TAM decisions, whereas the comparator remains TAM-off/TSA-only. For the first eligible trigger in rollout τ, we define

$$\widehat{\lambda}_{\tau,h} := \mathrm{MSE}_{\text{TSA-only},\tau,h} - \mathrm{MSE}_{\text{Full},\tau,h} .$$

Positive values indicate lower Full LTTS MSE, and negative values indicate higher Full LTTS MSE than the matched TAM-off branch. Because each pair shares its pre-trigger state and reference target, the contrast assesses subsequent prediction error; the Full LTTS branch continues its later TAM decisions. All 29,840 candidate patient–arm–seed rollouts satisfied the trigger and follow-up eligibility conditions. We evaluate each at four correlated offsets, yielding 119,360 comparisons.

We report offsetwise negative fractions of 5.9%, 7.5%, 9.6%, and 14.4% at $h = 1, 5, 10, 20$, respectively, with a reported pooled negative fraction of 9.4%. These fractions describe the closed-loop policy contrast at the four audited offsets.

I.5 TSA Replacement Backbones

We evaluate two TSA replacements, TimeKAN (Huang et al., 2025) and TimeMixer (Wang et al., 2024), preserving action routing, the output head, standardization, TAM interface, threshold policy, selective writeback, and recursive prediction. For each backbone, we use the same split, inputs, and 30 paired seeds for TAM-on/off; we fit TAM separately for each backbone using its paired checkpoint.

Table 10: **TSA replacement backbones.** We compare TimeKAN and TimeMixer with TAM-on/off across three benchmarks. Entries are mean ± sample SD over 30 paired seeds of per-seed equal-weight $1/3$ macro-averages of independently normalized metrics for both fixed continuations.

Backbone	TAM status	All-zero NMAE	All-zero NRMSE	All-one NMAE	All-one NRMSE
TimeKAN	TAM-on	0.322 ± 0.099	0.393 ± 0.104	0.418 ± 0.231	0.488 ± 0.235
TimeKAN	TAM-off	0.774 ± 0.106	0.843 ± 0.119	0.721 ± 0.137	0.913 ± 0.112
TimeMixer	TAM-on	0.383 ± 0.185	0.464 ± 0.210	0.264 ± 0.040	0.369 ± 0.049
TimeMixer	TAM-off	0.617 ± 0.104	0.745 ± 0.121	0.681 ± 0.078	0.784 ± 0.102

For both replacement backbones, we observe lower values in the TAM-on row than in its TAM-off counterpart in all four metrics; the TSA-full and TAM-off reference values are reported in Table 8.

J Reproducibility

J.1 Data, Selection, and Paired Evaluation

Appendices D and G fix dataset sizes, splits, settings, and window selection; synthetic reference branches share future innovations (Appendix D). We fix all preprocessing, model-selection, optimization, and rollout settings before testing, together with scoring and analysis rules. We access post-t_0 targets only for scoring. For each patient–arm call, we initialize a fresh private adapter with no state transfer across subjects, seeds, or continuations.

J.2 Action-Branch Training Evidence

In Algorithm 1, we filter source-only windows by action ($t = 11{:}50$, $l = 10$ for synthetic data; $t = 21{:}50$, $l = 20$ for MIMIC). Counts enumerate correlated transition positions. In teacher-forced validation, we measure conditional outcome-head error using the true next state.

Table 11: **Action-filtered TSA branch evidence.** In panel (a), we use TF NMAE with the true next state; in panel (b), we report PS TSA-only full-suffix NMAE. The matching PS frozen/adapted results are in Table 3a. Entries are mean ± sample SD over 30 seeds.

(a) Action-filtered teacher-forced diagnostic

Benchmark	l	n_0	n_1	N	$a = 1$ fraction	TF NMAE $a = 0$	TF NMAE $a = 1$
Linear	10	2,291	23,309	25,600	91.05%	0.182 ± 0.021	0.176 ± 0.019
Nonlinear	10	2,974	22,626	25,600	88.38%	0.236 ± 0.032	0.229 ± 0.030
MIMIC-derived	20	127,050	26,040	153,090	17.01%	0.271 ± 0.041	0.284 ± 0.044

(b) Predicted-state training diagnostic

Setting	PS TSA-only
Linear, $a = 0$	0.548 ± 0.073
Linear, $a = 1$	0.508 ± 0.066
Nonlinear, $a = 0$	0.536 ± 0.071
Nonlinear, $a = 1$	0.487 ± 0.063

J.3 Metrics and Aggregation

For all baseline, depth-wise, component, and backbone analyses, we use 30 paired seeds (0–29), with five seeds $\{0, 6, 12, 18, 24\}$ for window selection. We fix data generation and patient splits. Using separate random streams, we use seeds to control initialization, minibatch order, dropout, and

inference sampling. Across matched methods, we use the same data, partitions, prefixes, reference trajectories, interventions, and normalization statistics; internal random draws may differ across architectures. Sample SD uses $\mathrm{ddof} = 1$ and reflects this training/inference variation, conditional on the fixed data and split. For each seed, we pool errors over subjects and scored suffixes with training-prefix normalization. In Tables 8 and 10, we use per-dataset metrics, equal-weight $1/3$ macro-averages, square-rooted NRMSE, and mean $\pm$ sample SD over 30 macro-averages. We apply the same inclusion rules to all effect directions. For the post-trigger audit, we use ten seeds, 29,840 rollouts, and four correlated offsets (Appendix I.4); window tables use factual-prefix targets with the same aggregation.

J.4 HYPERPARAMETERS

Table 12: Core LTTS hyperparameters for the synthetic and MIMIC-IV-derived evaluations. Entries with two optimization values are listed as TSA; TAM.

Parameter	Synthetic	MIMIC-IV-derived
TSA model width	64	64
Attention heads	4	4
Offline learning rate (TSA; TAM)	10^{-3}; 10^{-3}	10^{-3}; 10^{-3}
Offline batch size (TSA; TAM)	256; 64	256; 64
Training epochs (TSA; TAM max.)	30; 50	30; 50
TAM early-stopping patience	5	5
Prototype count K	64	64
TAM monitoring-window cadence N'	10	10
TAM offline kernel-L_2 coefficient	10^{-5}	10^{-5}
Trigger threshold δ_{corr}	$\mu_{\mathrm{val}} + 3s_{\mathrm{val}}$	$\mu_{\mathrm{val}} + 3s_{\mathrm{val}}$
Private-adapter learning rate	5×10^{-4}	5×10^{-4}
Private-adapter update steps	5	5
Context length l	10	20