MULTIVARIATE TIME SERIES CAUSAL FOUNDATIONAL MODEL WITH IN-CONTEXT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting how a dynamic system evolves under a sequence of interventions is central to healthcare, economics, and engineering, and is complicated by time-varying confounding when treatments depend on history. Classical methods are trained from scratch per dataset and predict a single scalar outcome. Prior-data fitted networks (PFNs) enable training-free inference, but temporal PFNs either lack interventions or predict only one target variable, ignoring how the other covariates respond to treatment. We introduce MTS-CFM, a multivariate time series causal foundational model that forecasts every variable of the system, both the outcome and all covariates, over a multi-step horizon under a planned treatment sequence. Given an individual's history, a context set of related trajectories, and the treatment plan, it predicts the entire horizon in one forward pass without fine-tuning. MTS-CFM is trained entirely on synthetic trajectories sampled from a prior over temporal structural causal models with explicit treatment assignment. Without task-specific training, it performs competitively with domain-trained baselines and a causal PFN on interventional outcome prediction, while forecasting time-varying covariates more accurately than comparable baselines. These results make in-context learning a practical route to causal forecasting under dynamic treatments.

1 INTRODUCTION

Decision-making is often sequential, whether a clinician alters medication based on evolving vitals or an engineer adjusts a system in response to sensor feedback. The central challenge in these settings is predicting a system's trajectory under a specific sequence of interventions. Estimating this from observational data is difficult due to time-varying confounding. Because current treatments are determined by historical covariates, which were themselves influenced by past treatments, covariates act both as confounders and as mediators, and naive adjustment becomes biased (Robins, 1986; Robins et al., 2000). Accounting for this complex structure is the core of interventional prediction.

Existing estimators for this problem span from classical marginal structural models (Robins et al., 2000) to modern neural architectures utilizing recurrent networks (Lim, 2018; Li et al., 2021), balanced representations (Bica et al., 2020), and transformers (Melnychuk et al., 2022; Xiong et al., 2024; Hess et al., 2024). Despite this progress, these models must be trained from scratch for each new dataset, requiring substantial data, tuning, and computational resources. Furthermore, they in general restrict their predictions to a single scalar outcome. Real-world interventions, however, impact the entire system. A clinician, for instance, must monitor not just tumor size, but also blood counts and overall organ function, as these responses often determine the safety of a treatment.

Foundational models present a compelling alternative to training from scratch, leveraging in-context learning to rapidly adapt to unseen tasks (Bommasani et al., 2021; Brown et al., 2020). Prior-data fitted networks (PFNs) formalize this paradigm: by training a transformer exclusively on synthetic data from a prior distribution, the model learns to approximate Bayesian posterior inference in one forward pass (Müller et al., 2022). While PFNs and similar architectures have demonstrated impressive capabilities in tabular classification (Hollmann et al., 2025), zero-shot time-series forecasting (Dooley et al., 2023; Ansari et al., 2024; Das et al., 2024; Woo et al., 2024; Taga et al., 2025; Hoo et al., 2026), static causal inference (Balazadeh Meresht et al., 2026; Robertson et al., 2026; Ma et al., 2026), and even temporal contexts (Zare et al., 2026; Thumm & Chen, 2026), a critical gap remains. Standard temporal foundational models lack a mechanism for interventions; static causal

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

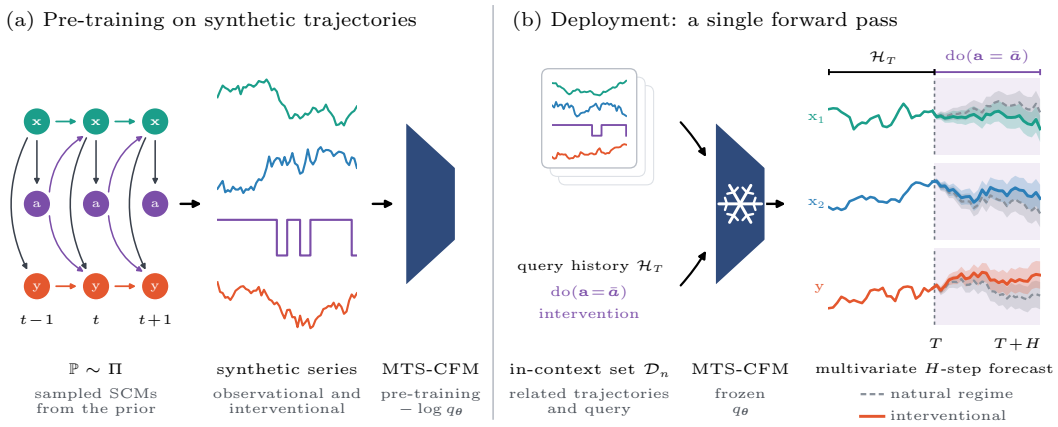

Figure 1: Overview of MTS-CFM. (a) The model learns entirely from synthetic multivariate trajectories drawn from a temporal SCM prior. (b) Using an in-context set and query history, MTS-CFM instantly forecasts all variates of the system under a planned intervention in a single forward pass.

PFNs cannot handle sequential treatments, and an existing longitudinal causal PFN is constrained to predicting a single target variable per time step, neglecting how the broader system responds.

We introduce MTS-CFM (illustrated in Figure 1), a multivariate time series causal foundational model that closes this gap. Given a unit's history (an individual trajectory, such as a single patient), a cohort of related trajectories, and a treatment plan specifying the intervention for the horizon, MTS-CFM predicts the future of the outcome and covariates in a single forward pass. It is trained entirely on synthetic data. On simulated benchmarks and a real-world dataset with time-varying covariates, it matches or improves upon domain-trained baselines and a longitudinal PFN on outcome prediction while also accurately forecasting all covariates in the system. Our contributions are as follows:

- We formalize multivariate causal forecasting under time-varying treatments as amortized interventional inference over a system's entire future, predicting every covariate and the outcome at each step of the horizon rather than a single scalar target per time step.
- We propose MTS-CFM, a transformer that attends sequentially across time, variables, and trajectories, supporting varying numbers of variables, time steps, and units at linear cost.
- We show that a single pre-trained model matches or exceeds specialized baselines and a longitudinal causal PFN on outcome prediction in systems with time-varying covariates, and provides the most accurate covariate forecasts among the comparable methods.

2 Related Work

PFNs. PFNs have emerged as a powerful paradigm for in-context learning. By pre-training on synthetic data drawn from a prior, PFNs perform inference on unseen tasks in a single forward pass without gradient updates. Müller et al. (2022) demonstrated that transformers can implicitly execute Bayesian inference when trained in this manner. Building on this foundation, Hollmann et al. (2025) introduced TabPFN, a tabular foundational model that solves classification and regression tasks in a fraction of a second while matching or exceeding traditional machine learning baselines.

Static causal PFNs. The in-context learning capabilities of PFNs have recently been directed towards causal effect estimation. For static cross-sectional data, Balazadeh Meresht et al. (2026) introduced CausalPFN to perform amortized causal effect estimation, while Robertson et al. (2026) proposed Do-PFN to leverage in-context learning for accurate causal inference. Ma et al. (2026) established the broader viability of foundational models for causal inference, introducing a principled structural causal model (SCM) prior framework that spans a range of causal inference settings, including back-door, front-door, and instrumental variable adjustment. However, these methods are limited to static settings and cannot handle time-varying treatments or temporal dynamics.

Time series models. Transformer architectures have significantly advanced time series forecasting. Nie et al. (2023) demonstrated that segmenting time series into patches improves forecasting performance by preserving local information. Subsequent architectures like Gateformer (Lan & Oermann, 2025) further advanced multivariate forecasting by introducing temporal and variate-wise attention combined with gated representations. The PFN framework has also been adapted for the temporal domain. Dooley et al. (2023) proposed ForecastPFN for synthetically trained zero-shot forecasting. Taga et al. (2025) extended these capabilities with TimePFN to improve multivariate forecasting via synthetic data generation, while Potapczynski et al. (2026) developed time-aware PFNs to incorporate exogenous variables into zero-shot forecasting pipelines. Bridging tabular and temporal modalities, Hoo et al. (2026) extended TabPFN to time series forecasting. All of these forecast observational futures, with no mechanism to condition on an intervention.

Time series causal inference. Estimating potential outcomes over time under dynamic treatment regimes began with Marginal Structural Models (Robins et al., 2000). To capture non-linear dynamics, Lim (2018) introduced Recurrent Marginal Structural Networks, and Li et al. (2021) proposed G-net, a recurrent approach to g-computation. Bica et al. (2020) later used adversarially balanced representations to adjust for time-varying confounding. Attention-based models followed, including the Causal Transformer (Melnychuk et al., 2022) and the G-Transformer (Xiong et al., 2024; Hess et al., 2024). Bridging the gap between temporal causal inference and in-context learning, Zare et al. (2026) explored causal longitudinal PFNs for interventional prediction, while Thumm & Chen (2026) studied interventional time series priors for causal PFNs. Both focus on a single target per step, whereas MTS-CFM forecasts the outcome and every covariate.

3 Problem Setup

3.1 Background on PFNs

PFNs for Static Data. Consider a supervised learning task with data-generating process $\mathbb{P}$ that belongs to some class of data-generating distributions $\mathcal{P}$. Rather than fitting a separate model to each observed dataset, PFNs treat the unknown $\mathbb{P}$ itself as a random object: a prior Π is placed over $\mathcal{P}$, and inference for $\mathbb{P}(\mathrm{y} \mid \mathbf{x} = \boldsymbol{x})$ is framed as Bayesian model averaging with respect to Π (Müller et al., 2022). A task, one dataset paired with its unknown data-generating process, is drawn by first sampling $\mathbb{P} \sim \Pi$ and then a context $\mathcal{D}_n = \{(\boldsymbol{x}_m, y_m)\}_{m=1}^n \overset{\text{i.i.d.}}{\sim} \mathbb{P}$, which updates the prior via Bayes' rule, $\Pi(\mathbb{P} \mid \mathcal{D}_n) \propto \Pi(\mathcal{D}_n \mid \mathbb{P})\,\Pi(\mathbb{P})$. Predictions at a query point $\boldsymbol{x}$ marginalize out $\mathbb{P}$:

$$\Pi(\mathrm{y} \mid \mathcal{D}_n, \boldsymbol{x}) = \int \mathbb{P}(\mathrm{y} \mid \boldsymbol{x})\,\Pi(\mathbb{P} \mid \mathcal{D}_n)\;\mathrm{d}\mathbb{P}.$$

This integral is generally intractable, so PFNs instead *amortize* it: a transformer $q_{\boldsymbol{\theta}}(\mathrm{y} \mid \mathcal{D}_n, \boldsymbol{x})$ is trained to map any context–query pair directly to an approximation of the posterior predictive, with attention aggregating information across the (order-invariant) context. The parameters $\boldsymbol{\theta}$ are fit once on synthetic episodes sampled from the prior, after which the network applies to any dataset well represented by the prior in a single forward pass, with no gradient updates. Training minimizes

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbb{P}\sim\Pi}\,\mathbb{E}_{\mathcal{D}_n,\,(\boldsymbol{x},y)\sim\mathbb{P}}\big[-\log q_{\boldsymbol{\theta}}(y \mid \mathcal{D}_n, \boldsymbol{x})\big].$$

Causal Inference for Static Data. An extension of the static setting introduces a treatment alongside covariates $\mathbf{x}$ and outcome y, so that the population is now $(\mathbf{x}, \mathrm{a}, \mathrm{y}) \sim \mathbb{P}$. Rather than the observational conditional $\mathbb{P}(\mathrm{y} \mid \mathbf{x} = \boldsymbol{x}, \mathrm{a} = a)$, the target of inference is the *conditional interventional distribution* (CID) $\mathbb{P}(\mathrm{y} \mid \mathbf{x} = \boldsymbol{x}, \mathrm{do}(\mathrm{a} = a))$, obtained by intervening on the treatment via Pearl's do-operator (Pearl, 2009). Building on the PFN framework above, recent work amortizes this interventional query itself: a prior Π is placed over SCMs rather than over $\mathbb{P}$ directly, the context $\mathcal{D}_n = \{(\boldsymbol{x}_m, a_m, y_m)\}_{m=1}^n$ is sampled from the induced observational distribution, the query outcome is sampled from the interventional distribution obtained by applying $\mathrm{do}(\mathrm{a} = a)$ to the sampled SCM, and a transformer $q_{\boldsymbol{\theta}}\big(\mathrm{y} \mid \mathcal{D}_n, \boldsymbol{x}, \mathrm{do}(\mathrm{a} = a)\big)$ approximates the CID (Robertson et al., 2026).

Time series PFNs. Static PFNs treat $(\boldsymbol{x}_m, y_m)$ as exchangeable draws from a fixed $\mathbb{P}$, which ignores the sequential dependence of time series. Temporal PFNs instead place a prior Π over

stochastic processes, sampling whole time series rather than i.i.d. tuples. Given an observed series $\boldsymbol{X}_{0:T} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_T)$, the goal is defined over a future horizon H as

$$\Pi(\mathbf{X}_{T+1:T+H} \mid \boldsymbol{X}_{0:T}) = \int \mathbb{P}(\mathbf{X}_{T+1:T+H} \mid \boldsymbol{X}_{0:T})\, \Pi(\mathbb{P} \mid \boldsymbol{X}_{0:T})\, \mathrm{d}\mathbb{P},$$

which a probabilistic temporal PFN $q_{\boldsymbol{\theta}}(\mathbf{X}_{T+1:T+H} \mid \boldsymbol{X}_{0:T})$ can approximate by minimizing the analogous objective $\mathbb{E}_{\mathbb{P}\sim\Pi}\, \mathbb{E}_{\boldsymbol{X}_{0:T+H}\sim\mathbb{P}}\big[-\log q_{\boldsymbol{\theta}}(\boldsymbol{X}_{T+1:T+H} \mid \boldsymbol{X}_{0:T})\big]$. Existing temporal PFNs differ in how they model this target: some produce point forecasts (Dooley et al., 2023; Taga et al., 2025), and some reuse tabular PFNs with time-derived features (Hoo et al., 2026).

3.2 Problem Formulation

The two settings above address orthogonal limitations: temporal PFNs forecast future values without interventions, while causal PFNs support interventional queries only for static data. We consider the setting that combines both. At each time t, we observe a covariate $\mathbf{x}_t$, treatment a_t, and outcome y_t, and treatments may depend on the evolving history, which induces time-varying confounding. We index time from zero, so a unit observed for $T+1$ steps has its last observation at index T.

Rather than restricting attention to a single scalar outcome, we treat the observed system as inherently multivariate. At each t, the covariate $\mathbf{x}_t \in \mathbb{R}^{d_x}$ is itself vector-valued, describing the full evolving state of the system alongside the treatment a_t, while the outcome $\mathrm{y}_t \in \mathbb{R}$ remains a scalar quantity of interest, so that the object to be forecast under intervention is the *entire future covariate system together with the outcome*, rather than the outcome alone. Formally, let $\mathrm{x}_{m,i,t}$ denote the scalar variate i of sample m at time t. For brevity, we drop indices when the context is unambiguous, writing the vector $\mathbf{x}_t$ for the full covariate state at time t, or x_i when isolating a specific scalar variate. Each sample additionally carries a vector of static covariates $\mathbf{s} \in \mathbb{R}^{d_s}$ which may affect both the treatment assignment and the dynamics and therefore is part of the conditioning set rather than of the forecast target. The treatment is categorical, $\mathrm{a}_t \in \mathcal{A} = \{0, 1, \ldots, n_a - 1\}$.

The prediction is informed by an in-context dataset $\mathcal{D}_n = \{\mathcal{T}_m\}_{m=1}^{n}$ of n trajectories $\mathcal{T}_m = (\boldsymbol{s}_m, \boldsymbol{X}_{m,0:T_m}, \boldsymbol{a}_{m,0:T_m-1}, \boldsymbol{y}_{m,0:T_m})$ drawn i.i.d. from $\mathbb{P}$, whose lengths T_m may differ from one another and from the query's. Given the query unit's static covariates $\boldsymbol{s}$ and its own observed history $\mathcal{H}_T = (\boldsymbol{x}_0, y_0, a_0, \ldots, a_{T-1}, \boldsymbol{x}_T, y_T)$, the in-context dataset $\mathcal{D}_n$, and a treatment plan $\bar{\boldsymbol{a}} = (\bar{a}_T, \ldots, \bar{a}_{T+H-1}) \in \mathcal{A}^H$ applied over the window $T{:}T{+}H{-}1$ with horizon H, the goal is to estimate the CID of the *entire* future system state and outcome over $T{+}1{:}T{+}H$ under the planned intervention $\mathrm{do}(\mathbf{a}_{T:T+H-1} = \bar{\boldsymbol{a}})$, abbreviated $\mathrm{do}(\mathbf{a} = \bar{\boldsymbol{a}})$,

$$\begin{aligned}\Pi\big((\mathbf{X}, \mathbf{y})_{T+1:T+H} \mid \boldsymbol{s}, \mathcal{H}_T, \mathcal{D}_n, \mathrm{do}(\mathbf{a} = \bar{\boldsymbol{a}})\big) = \int \mathbb{P}\big((\mathbf{X}, \mathbf{y})_{T+1:T+H} \mid \boldsymbol{s}, \mathcal{H}_T, \mathrm{do}(\mathbf{a} = \bar{\boldsymbol{a}})\big) \\ \times\, \Pi(\mathbb{P} \mid \boldsymbol{s}, \mathcal{H}_T, \mathcal{D}_n)\, \mathrm{d}\mathbb{P},\end{aligned} \tag{1}$$

where $\mathbf{X}_{T+1:T+H} \in \mathbb{R}^{H \times d_x}$ and $\mathbf{y}_{T+1:T+H} \in \mathbb{R}^H$ denote the covariate and outcome trajectories over the full forecast horizon. Note that the plan is indexed $T{:}T{+}H{-}1$ while the forecast target is indexed $T{+}1{:}T{+}H$. The treatment $\bar{a}_t$ is the one that drives the transition into the state at $t+1$, so the two windows are offset by one step and the plan contains no treatment beyond the last predicted state. Amortizing this quantity requires a network

$$q_{\boldsymbol{\theta}}\big((\mathbf{X}, \mathbf{y})_{T+1:T+H} \mid \boldsymbol{s}, \mathcal{H}_T, \mathcal{D}_n, \mathrm{do}(\mathbf{a} = \bar{\boldsymbol{a}})\big), \tag{2}$$

trained on multivariate trajectories generated under an explicit treatment-assignment mechanism depending on both the static covariates and the evolving history, so that both observational and interventional regimes, and the dependencies between system variables, are represented in the prior.

Identifiability. Equation (1) is a posterior predictive, not a point estimate, so the usual causal assumptions enter through the prior Π rather than through an adjustment argument on the target dataset: the objective is well-posed as long as every $\mathbb{P}$ meets the four conditions for the g-formula (Robins, 1986), which Π satisfies by construction (Appendix B.2). Sequential ignorability, the condition that treatments depend only on the history $\mathcal{H}_t$ and statics $\boldsymbol{s}$, still allows time-varying confounding and latent traits that act on the future through the observed trajectory; it rules out only a hidden cause of both treatment and outcome. Nothing further is required of the target dataset. Yet, as for any PFN, the guarantee is inherited only when the dataset falls within the support of Π.

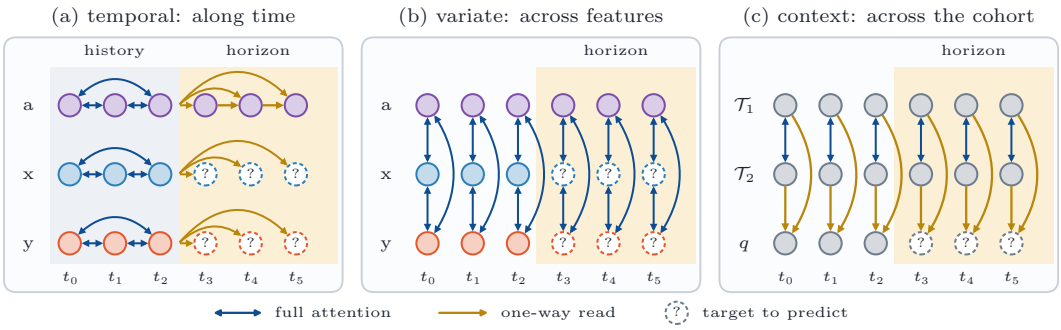

Figure 2: Each layer applies three axis-wise attentions in sequence: (a) temporal, within a variate over time, where targets read the history but are never read back; (b) variate, full across treatment, covariates and outcome at one step; (c) context, a one-way read from the cohort to the query. Context attention comes last, once the query has absorbed its history and variate–treatment interactions.

4 MTS-CFM

4.1 Architecture

Token grid. MTS-CFM structures each task as a three-dimensional token grid indexed by (sample, time, variate). The sample dimension contains the n in-context trajectories alongside the target query sample, while the time dimension covers the interval $0{:}T{+}H$ and any longer in-context trajectory. The variate dimension concatenates the outcome and the d_x covariates, treating them as channels that share one input projection within a unified multivariate state that is processed by a single predictive head. Each cell in this grid is projected to $\mathbb{R}^{d_{\text{model}}}$ via a shared linear transformation. Distinct identity of each variate is instead provided by an additive learned embedding (Appendix A.1).

Treatments are one-hot encoded, linearly projected to $\mathbb{R}^{d_{\text{model}}}$, and added as an additional channel along the variate axis. This treatment sequence is temporally shifted such that the slice at time t contains a_{t-1}, reflecting the treatment that induced the transition to the state $(\mathbf{x}_t, \mathrm{y}_t)$. Each token is subsequently augmented with two additional features: a learned unit-type indicator to distinguish in-context trajectories from the query, and an MLP-derived embedding of the unit's static covariates $\boldsymbol{s}$. For time steps beyond T, the unobserved query cells are populated with a learned mask token.

Transformer layer. Each of the L layers applies three attention operations along a single axis of the grid, the other two folded into batch (Figure 2): *temporal* attention over time at fixed sample and variate, which learns each channel's dynamics and carries rotary positional embeddings (RoPE) (Su et al., 2024); *variate* attention over the $2 + d_x$ rows at fixed sample and time, which learns channel interactions and couples the treatment into the state; and *context* attention over the $n + 1$ units at fixed time and variate, the in-context learning axis as in TabPFN (Hollmann et al., 2025). Pairing cells at a common time index takes the cohort to share a clock origin, such as treatment initiation or admission. Each is a pre-norm residual block, and a pre-norm feed-forward block closes the layer.

Analogous to axial attention (Ho et al., 2019), the sequential composition of the three sub-blocks ensures that any pair of cells can exchange information within a single layer, subject to the masks. Beginning with the second layer, each attention operation processes representations that have already been fused across the other two dimensions. Thus, the three mechanisms are deeply coupled rather than operating in parallel. Specifically, temporal attention processes a sequence where each time step already encapsulates the variate system and the broader cohort. Similarly, context attention compares units based on states that fully integrate observed trajectories and applied treatments. This encoding is precisely what enables the cohort to provide information about a counterfactual rather than simply a marginal distribution. As a result, deeper layers successfully capture the complex interactions between all three factors, achieving this at a computational cost that scales with the sum rather than the product of the axis lengths. Placing context attention last ensures the query interrogates the cohort only after absorbing its own history, characteristics, and treatment plan.

Masking. An observation mask hides unobserved cells, padded variates, and padded units from all three attention operations, which lets the model handle varying d_x, T_m, and n. The query's horizon cells are treated differently: they hold a learned mask embedding and exchange information with the other variates at the same time step, including the planned treatment, but are never read by cells at other time steps, so no unknown future state serves as temporal context. Otherwise, temporal attention is bidirectional over observed data: in-context units read their entire trajectories, and the query reads its full history, though its history tokens never read the plan. Within the horizon, only the treatment row is observed, since the plan $\bar{\boldsymbol{a}}$ is known, and it is read causally, so the treatment token at $T{+}h$ summarizes the observed treatment history and the planned treatments up to $\bar{a}_{T+h-1}$, but no later one; the masked state cells read the history alone. The prediction at $T{+}h$ thus never sees a later treatment, and all H steps are predicted in one forward pass without feeding predictions back across time, though horizon cells still interact within a time step through variate attention. In context attention, the query reads all trajectories in $\mathcal{D}_n$, whereas in-context units read only one another, so no information from the query leaks into the cohort. The full masks for all units, including the in-context trajectories, are shown per axis in Figures 3, 4, and 5, with further details in Appendix A.3.

Prediction head and objective. After the final layer, the query's horizon tokens are extracted, and the treatment channel is discarded. A shared predictive head then projects each remaining token into a univariate Gaussian mixture model with K components. This factorizes the network of Equation (2) as $q_{\boldsymbol{\theta}}\big((\mathbf{X},\mathbf{y})_{T+1:T+H} \mid \cdot\big) = \prod_{h=1}^{H} q_{\boldsymbol{\theta}}\big(\mathrm{y}_{T+h} \mid \cdot\big) \prod_{i=1}^{d_x} q_{\boldsymbol{\theta}}\big(\mathrm{x}_{i,T+h} \mid \cdot\big)$, where $\cdot$ denotes the conditioning set of Equation (2), so the head recovers the per-variable, per-step marginals of the CID. The factors are not independent predictions. Within the horizon, the treatment row is read causally, so the token at step $T{+}h$ carries the history together with the preceding planned treatments, and variate attention then injects this summary into every target variate at that step. The outcome and all covariates at a step are therefore built from the same conditioning, and the dependence between variables is carried by these tokens rather than by the factorization. The training objective minimizes the negative log-likelihood (NLL) of this mixture over all valid cells, plus two regularization terms: a Huber loss applied to the mixture mean, and a hinge penalty on the maximum mixture weight. The Huber loss is introduced because NLL alone can be reduced by widening the predictive distribution rather than improving its mean, which can degrade the point estimates used during evaluation. Meanwhile, the hinge penalty prevents the distribution from collapsing into a single Gaussian mode (Zare et al., 2026). More details are provided in Appendix A.4.

4.2 Training

Prior. MTS-CFM is trained exclusively on synthetic data from the temporal SCM prior of Zare et al. (2026), whose explicit treatment-assignment mechanism provides the interventional targets required by our objective. We summarize it here and give full details in Appendix B. Each task draws a fresh $\mathbb{P} \sim \Pi$, used for one gradient update and then discarded. The generator first samples the dimensionality $d_x \sim \mathcal{U}\{1,\dots,10\}$ and maximum lag $K \in \{1,2\}$, then a sparse acyclic graph for instantaneous relationships and lagged dependencies whose density decays geometrically with lag. Each variable receives its own noise profile, a simple non-linear function, and a persistence term, giving a mix of deterministic, stochastic, smooth, and non-smooth dynamics. For a random subset of covariates, this mechanism is replaced by one of five motifs: leaky action memory, saturating accumulation, homeostatic reversion, error-driven feedback, or a smoothed readout of another variable. These motifs mimic regulated, bounded, or delayed responses and are the only pathways through which interventions alter the covariate state. The outcome is a separate autoregressive process that tracks either a single variate or a hidden affine projection of the entire state, with its own noise, a weak trend, and both direct and accumulated treatment effects. A per-unit latent confounder $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_3)$ drives both the initial state and the treatment policy; in 30% of tasks the model also observes static covariates $\mathbf{s}$, noisy and partially discretized proxies for $\mathbf{z}$. Treatments follow a stochastic behavior policy driven by the current state, treatment history, and $\mathbf{z}$. Because the policy reacts to states altered by earlier treatments, the trajectories exhibit time-varying confounding.

Datasets and targets. Each training dataset contains a cohort of $n \sim \mathcal{U}\{3,\dots,200\}$ samples alongside a single query sample from the same task. All samples are simulated forward under the behavior policy, the query until a random cutoff time $T \sim \mathcal{U}\{1,\dots,60\}$ and each in-context sample until its own, separately sampled length. For 70% of these datasets, an intervention plan $\bar{\boldsymbol{a}}$

is generated for the future window $T{:}T{+}H{-}1$, and the query sample is re-simulated under this new plan, keeping its latent confounder $\mathbf{z}$, mechanisms, and $\mathcal{H}_T$ fixed. In the remaining 30%, the query unit continues under the behavior policy to produce a factual target, ensuring that standard forecasting remains within the training distribution. The supervised target is thus a draw from $\mathbb{P}\big((\mathbf{X}, \mathbf{y})_{T+1:T+H} \mid \boldsymbol{s}, \mathcal{H}_T, \mathrm{do}(\mathbf{a} = \bar{\boldsymbol{a}})\big)$, covering every variable over the horizon rather than only the outcome at one time step. To prevent information leakage, inputs and labels are normalized using statistics computed from the cohort, never incorporating the query's future data.

Optimization. Because episodes are generated on the fly, the model never encounters the same data twice, removing the need for a training and validation split over the prior. A single optimization step processes $B = 256$ newly sampled tasks, divided into 64 micro-batches for gradient accumulation. Since peak activation memory depends only on the micro-batch size, accumulation allows large cohorts and long time axes without exceeding memory limits. Each micro-batch contributes to the gradient in proportion to its weighted number of supervised cells, $\sum_c w_c$ in Equation (8) in Appendix A.4, to avoid mean-of-means averaging across episodes with different d_x. Training runs for 10,000 steps, totaling 2.56 million unique synthetic episodes. We use the AdamW (Loshchilov & Hutter, 2019) optimizer with a linear learning rate warmup and subsequent cosine decay. We use global-norm gradient clipping and the selective activation checkpointing described in Appendix A.5. All hyperparameters are in Tables 4 and 5, and the prior's sampling ranges are in Appendix B.

5 Experiments

5.1 Setup

To assess our approach, we use four domains: cancer tumor growth (Geng et al., 2017), HIV treatment dynamics (Adams et al., 2004), pharmacokinetic/pharmacodynamic (PK/PD) models of warfarin (Holford, 1986; Consortium, 2009; Hamberg et al., 2010), and intensive care records from MIMIC-III (Johnson et al., 2016; Pollard et al., 2026; Wang et al., 2020). Full technical details for these environments are available in Appendix C. Since the cancer, warfarin, and HIV domains are driven by synthetic, partially mechanistic models, they support interventional forward simulations. We can therefore compute exact interventional values by replaying a patient's dynamics under any treatment sequence. By contrast, MIMIC-III consists entirely of retrospective, real-world patient logs. Hence, we use it only to assess factual predictions under the care that was administered. We follow the evaluation protocol used in Zare et al. (2026), casting all domains into a common time-series format, where patient histories are capped at 60 steps and a horizon of $H = 5$. For each domain, a task fixes a confounding level γ (ten levels), a context size ($n \in \{40, 80, 160, 320, 500\}$) and one of two repetitions with different random seeds, giving $10 \times 5 \times 2 = 100$ tasks. In the simulated domains, γ controls the strength of time-varying confounding. In MIMIC-III, confounding cannot be varied, so its tasks instead differ in which patients are randomly drawn as context and query. A query asks for a patient's future under one treatment plan from one forecast origin T. Every time step from $T = 10$ onward serves as an origin, and each query uses only the observed history up to that point. At each origin, the simulators provide one-step queries under all four treatments and horizon-5 queries under ten plans sampled from $\mathcal{A}^H$, whereas MIMIC-III provides a single factual query of each kind. Appendix C gives the number of queries per domain and task.

Baselines. We compare MTS-CFM with seven baselines: six dataset-specific estimators for longitudinal treatment effects, covering the main methodological families, and one longitudinal causal PFN. The marginal structural model (MSM) (Robins et al., 2000) and Recurrent Marginal Structural Networks (RMSN) (Lim, 2018) correct for time-varying confounding through inverse-probability weighting. G-Net (Li et al., 2021) performs g-computation with recurrent networks. Counterfactual Recurrent Networks (CRN) (Bica et al., 2020) learn treatment-invariant representations through adversarial balancing. The Causal Transformer (CT) (Melnychuk et al., 2022) and G-Transformer (GT) (Hess et al., 2024) use attention-based sequence models. These six select hyperparameters on a validation split of the context trajectories, then fit final weights on the full context set before the queries are scored. The seventh baseline, CausalLongPFN (CLPFN) (Zare et al., 2026), is, like MTS-CFM, pre-trained only on synthetic data and receives the context trajectories as in-context input, with no gradient updates at test time. Unlike MTS-CFM, it predicts a single target variable at a time rather than the full system. For outcome prediction, we report all seven baselines as published

Table 1: Outcome prediction results. Normalized RMSE, lower is better. **Bold**, underline and *italic* mark the best, second-best and third-best result per column.

	Cancer ($d_x = 0$)		HIV ($d_x = 5$)		MIMIC-III ($d_x = 9$)		Warfarin ($d_x = 9$)	
Model	One-step	Horizon-5	One-step	Horizon-5	One-step	Horizon-5	One-step	Horizon-5
MTS-CFM	**0.162±0.246**	*0.302±0.361*	**0.042±0.016**	**0.159±0.091**	**0.592±0.242**	**0.676±0.293**	0.032±0.023	0.168±0.131
CLPFN	*0.167±0.255*	0.385±0.356	0.066±0.032	0.288±0.174	0.617±0.256	0.688±0.198	*0.036±0.023*	*0.196±0.143*
MSM	0.200±0.278	0.465±0.435	*0.061±0.027*	0.248±0.122	0.619±0.256	0.809±0.260	**0.014±0.007**	**0.152±0.075**
RMSN	0.166±0.256	**0.246±0.337**	0.051±0.029	0.186±0.117	0.626±0.272	0.729±0.226	0.055±0.085	0.238±0.236
G-Net	0.168±0.242	0.308±0.285	0.097±0.066	*0.235±0.137*	*0.619±0.246*	*0.710±0.193*	0.102±0.109	0.261±0.213
CRN	0.251±0.291	0.278±0.334	0.244±0.175	0.405±0.253	0.622±0.249	0.725±0.214	0.270±0.191	0.480±0.315
CT	0.209±0.265	0.849±0.743	0.100±0.058	0.915±0.618	0.638±0.264	0.972±0.373	0.084±0.091	0.749±0.615
GT	0.217±0.281	0.372±0.456	0.094±0.056	0.342±0.193	0.620±0.251	0.694±0.196	0.158±0.188	0.546±0.646

by Zare et al. (2026) and evaluate MTS-CFM with their released code and protocol, so the numbers refer to the same domains, task grid, and queries. The covariate forecasting results are our own.

Prediction protocol. Every method sees the same information for each query: the static covariates $\boldsymbol{s}$, the unit's history $\mathcal{H}_T$ up to the forecast origin T, the context set $\mathcal{D}_n$, and the planned treatments $\bar{\boldsymbol{a}} = (\bar{a}_T, \ldots, \bar{a}_{T+H-1})$. The first planned treatment $\bar{a}_T$ produces the state at $T+1$, and a prediction at step $T+h$ may use only $\bar{a}_T, \ldots, \bar{a}_{T+h-1}$. We report one-step ($h = 1$) and horizon-5 ($h = H = 5$) performance. In the cancer, warfarin, and HIV simulators, the plan is an intervention, and the ground truth is obtained by running simulator forward from the unit's state at T under that plan. In MIMIC-III, the plan is the plan actually administered, and the ground truth is the observed future. MTS-CFM returns all H steps in one forward pass, and its point forecast is the mean of its predictive distribution.

Covariate forecasting. Most longitudinal estimators forecast only the outcome, so the multivariate forecasts of MTS-CFM can be compared against only a few methods. G-Net predicts covariates natively: its g-computation rollout forecasts every covariate at each future step. CLPFN, in contrast, forecasts one target column and cannot predict multivariate targets. We therefore query it once per covariate, labelling covariate j as the target variable and the original outcome as one of the remaining variates, which leaves the conditioning set unchanged and only alters which variable is forecast (Appendix D.1). This needs d_x separate runs per task, each predicting a single covariate, whereas MTS-CFM forecasts all covariates and the outcome in a single pass. Covariate accuracy is computed per variable on the normalized scale and averaged over the d_x covariates of a domain. Because the cancer benchmark has no covariates, this comparison covers HIV, warfarin, and MIMIC-III only.

Metrics. We report root mean square error (RMSE) on a normalized scale. Normalization statistics are computed from the context trajectories alone. For each task, we pool the errors of all scored queries, average the task-level values within a domain, and report the standard deviation. Because both PFNs output predictive distributions, we also compare their probabilistic quality. The metrics are NLL, the continuous ranked probability score (CRPS), calibration error of the probability integral transform (PIT-ECE), and the coverage and width of central 80% and 90% prediction intervals.

5.2 Results

Outcome prediction. Without any gradient updates, MTS-CFM is competitive with estimators retrained on every task (Table 1). It is best at both horizons on HIV, where it reduces the horizon-5 RMSE of CLPFN by 45%, and on MIMIC-III. On warfarin, it improves on CLPFN at both horizons and is second to MSM. Warfarin's INR-based dosing and near-linear one-step dynamics suit the MSM, whose margin fades by horizon-5. Cancer is an outlier for our setting with no time-varying covariate; variate attention has nothing to exploit and MTS-CFM reduces to an outcome-only forecaster. It is still the most accurate at one step but trails RMSN and CRN at horizon-5.

Covariate prediction. MTS-CFM has the lowest covariate error in four of six settings, ties CLPFN on one-step warfarin (Table 2), and loses only on one-step MIMIC-III. Its advantage grows with the horizon, with reductions of 39% (HIV), 31% (warfarin), and 6% (MIMIC-III) relative to CLPFN at horizon-5. It is best on 20 of 23 covariates at horizon-5 and 16 of 23 at one step (Tables 10–12 in Appendix E). Single-target models are competitive for one-step regression, but

Table 2: Covariate prediction results. Normalized RMSE averaged over covariates, lower is better. Per-covariate results provided in Tables 10-12 (Appendix E).

	HIV ($d_x = 5$)		MIMIC-III ($d_x = 9$)		Warfarin ($d_x = 9$)	
Model	One-step	Horizon-5	One-step	Horizon-5	One-step	Horizon-5
MTS-CFM	**0.032±0.020**	**0.149±0.096**	0.354±0.073	**0.434±0.092**	**0.026±0.005**	**0.095±0.020**
CLPFN	0.047±0.025	0.245±0.141	**0.335±0.091**	0.460±0.105	**0.026±0.010**	0.137±0.042
G-Net	0.087±0.049	0.232±0.121	0.383±0.112	0.503±0.114	0.131±0.139	0.231±0.140

Table 3: One-step probabilistic prediction quality of the two PFNs, scored on the outcome variable only. Coverage is better the closer it is to its nominal level.

Domain	Model	RMSE ↓	NLL ↓	CRPS ↓	PIT-ECE ↓	Cov. 80%	Width 80%	Cov. 90%	Width 90%
Cancer	MTS-CFM	0.162	-0.876	0.081	0.028	70.3%	0.127	77.0%	0.169
	CLPFN	0.167	-0.711	0.082	0.029	70.3%	0.125	78.1%	0.170
HIV	MTS-CFM	0.042	-1.670	0.025	0.024	68.4%	0.081	79.0%	0.105
	CLPFN	0.066	-1.310	0.039	0.037	75.3%	0.155	84.9%	0.205
MIMIC-III	MTS-CFM	0.592	0.840	0.321	0.008	81.6%	1.376	90.0%	1.809
	CLPFN	0.617	0.938	0.336	0.019	72.7%	1.096	83.6%	1.518
Warfarin	MTS-CFM	0.032	-2.154	0.018	0.047	80.4%	0.074	87.5%	0.097
	CLPFN	0.036	-1.976	0.021	0.036	86.3%	0.106	93.4%	0.145

multi-step forecasts benefit from modeling how variables evolve together under the plan, which MTS-CFM does in one pass instead of d_x. G-Net's autoregressive rollout feeds its own predictions back as inputs, so errors compound across steps even on near-constant covariates such as age.

Calibration. Table 3 compares the two PFNs' one-step predictive distributions for the outcome; MTS-CFM is scored on the same scalar marginal as CLPFN. MTS-CFM achieves lower NLL and CRPS on all four domains. The clearest calibration differences are on HIV and MIMIC-III, where MTS-CFM cuts CLPFN's PIT-ECE by a third and by more than half. On HIV, its intervals are much sharper, roughly half as wide, at lower coverage. On MIMIC-III, its intervals are wider, but their coverage is closer to the nominal levels. On cancer, the two models perform similarly, and on warfarin MTS-CFM is more accurate and sharper at a slightly higher PIT calibration error. Coverage depends more on the domain than on the model: on cancer and HIV both models fall below the nominal levels, while on warfarin CLPFN exceeds them and MTS-CFM stays close.

6 Conclusion

We introduced MTS-CFM, a multivariate time series causal foundational model that predicts the interventional distribution of a system's entire future, including both outcome and covariates. MTS-CFM is trained only on synthetic trajectories from a prior over temporal SCMs, and it answers a new query in a single forward pass by attending in turn across time, variables, and in-context trajectories. On benchmarks with time-varying covariates, the same pre-trained model matches or improves upon baselines trained from scratch on each dataset. Our model also gives the most accurate covariate forecasts in settings we evaluated, and yields better-calibrated predictive distributions on most domains. These findings indicate that in-context learning over a sufficiently rich causal prior is a practical route to training-free forecasting of how a whole system responds to treatment, which is closer to what sequential decision-making in medicine and other fields actually requires.

Limitations and future work. MTS-CFM is only as reliable as the match between its prior and the target data. Larger systems, continuous doses, and time-varying unobserved confounding lie outside the prior, and the model does not flag such queries. On covariate-free systems such as cancer, it offers no structural advantage. The cohort is indexed on a common clock, so units with staggered entry or irregular sampling must first be aligned to a shared origin. Interventional performance is evaluated on simulations only. Finally, the head predicts marginal mixtures, so it cannot sample coherent joint trajectories. Future work includes broader priors and joint predictive distributions.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on the design of our experiments, to assist in implementing parts of our method, to reformat the benchmark datasets into the input format required by MTS-CFM, and to help interpret results. Specifically, AI assistance was used to implement the selective activation checkpointing scheme (Appendix A.5), including identifying which computationally heavy operations to retain. The dataset reformatting only restructured existing benchmark data. The data-generating processes and benchmark generation code of Zare et al. (2026) were not altered. We have not used generative AI tools to generate synthetic or benchmark data, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide ingredients for or write mathematical proofs, or propose or refine research hypotheses. Translation and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to draft parts of the paper, to edit the paper to improve readability, to summarize existing literature, to propose keywords, and to write the Python scripts used to generate the figures. We have reviewed all AI-assisted work. All AI-assisted code, including the checkpointing implementation and the figure scripts, was reviewed and tested by the authors. All AI-generated literature summaries were verified, and all cited works were checked. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses one real-world dataset, MIMIC-III (Johnson et al., 2016; Pollard et al., 2026), which consists of de-identified intensive care records. We accessed it through PhysioNet under its credentialed data use agreement and followed its terms, including the prohibition on attempting to re-identify patients. The remaining three domains are synthetic or mechanistic simulators and involve no human subjects. MTS-CFM is intended as a research contribution and is not validated for clinical use: MIMIC-III supports only factual prediction, interventional accuracy is assessed solely on simulators, and the model gives no warning when a query falls outside its prior. Any deployment in medicine or another high-stakes setting would require domain-specific validation and supervision.

REPRODUCIBILITY STATEMENT

We have taken several steps to make this work reproducible. Section 4.1 and Appendix A describe the MTS-CFM architecture, including tokenization, the three attention axes and their masks (Appendix A.3), the prediction head and training objective (Appendix A.4, Eq. 8), and the memory-saving scheme (Appendix A.5). Table 4 lists all architecture hyperparameters. Appendix B specifies the synthetic prior in full, including graph sampling, the generic mechanism, the dynamical motifs (Table 6), the behaviour policy, and the outcome process. Section 4.2 and Table 5 give the episode construction and optimization settings. For evaluation, we use the four benchmark domains, data-generating code, and protocol of Zare et al. (2026) without modification. Appendix C summarizes these domains, and Appendix D describes the baselines, their search grids (Table 9), and how CLPFN is queried per covariate for covariate forecasting (Appendix D.1). All results are aggregated over the same grid of context sizes, confounding levels, and random seeds (Section 5.1). The CLPFN comparison uses its publicly released checkpoint. MIMIC-III is available through PhysioNet to credentialed users, and we follow the preprocessing described in Appendix C.4. Upon acceptance, we will publicly release the source code together with the pre-trained model weights.

REFERENCES

Brian Michael Adams, Harvey Thomas Banks, Hee-Dae Kwon, and Hien T Tran. Dynamic multidrug therapies for hiv: Optimal and sti control approaches. Technical report, North Carolina State University. Center for Research in Scientific Computation, 2004.

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=gerNCVqqtR. Expert Certification.

Vahid Balazadeh Meresht, Hamidreza Kamkari, Valentin Thomas, Junwei Ma, Bingru Li, Jesse Cresswell, and Rahul Krishnan. Causalpfn: Amortized causal effect estimation via in-context learning. *Advances in Neural Information Processing Systems*, 38:154945–154984, 2026.

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*, 2020. URL https://openreview.net/forum?id=BJg866NFvB.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

International Warfarin Pharmacogenetics Consortium. Estimation of the warfarin dose with clinical and pharmacogenetic data. *New England Journal of Medicine*, 360(8):753–764, 2009.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=jn2iTJas6h.

Samuel Dooley, Gurnoor Singh Khurana, Chirag Mohapatra, Siddartha V Naidu, and Colin White. Forecastpfn: Synthetically-trained zero-shot forecasting. *Advances in Neural Information Processing Systems*, 36:2403–2426, 2023.

Changran Geng, Harald Paganetti, and Clemens Grassberger. Prediction of treatment response for combined chemo-and radiation therapy for non-small cell lung cancer patients using a biomathematical model. *Scientific reports*, 7(1):13542, 2017.

A-K Hamberg, Mia Wadelius, Jonatan D Lindh, Marja-Liisa Dahl, Roberto Padrini, Panos Deloukas, Anders Rane, and E Niclas Jonsson. A pharmacometric model describing the relationship between warfarin dose and inr response with respect to variations in cyp2c9, vkorc1, and age. *Clinical Pharmacology & Therapeutics*, 87(6):727–734, 2010.

Konstantin Hess, Dennis Frauen, Valentyn Melnychuk, and Stefan Feuerriegel. G-transformer for conditional average potential outcome estimation over time. *arXiv preprint arXiv:2405.21012*, 2024.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

Nicholas HG Holford. Clinical pharmacokinetics and pharmacodynamics of warfarin: understanding the dose-effect relationship. *Clinical pharmacokinetics*, 11(6):483–504, 1986.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. From tables to time: Extending tabPFN-v2 to time series forecasting. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=KIkQj8VOUY.

Alistair Johnson, Tom Pollard, and Roger Mark. MIMIC-III Clinical Database. *PhysioNet*, September 2016. doi: 10.13026/C2XW26. URL https://doi.org/10.13026/C2XW26. Version 1.4.

Vijay Anand Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing activation recomputation in large transformer models. *Proceedings of Machine Learning and Systems*, 5:341–353, 2023.

Yu-Hsiang Lan and Eric K Oermann. Gateformer: Advancing multivariate time series forecasting through temporal and variate-wise attention with gated representations. *arXiv preprint arXiv:2505.00307*, 2025.

Rui Li, Stephanie Hu, Mingyu Lu, Yuria Utsumi, Prithwish Chakraborty, Daby M Sow, Piyush Madan, Jun Li, Mohamed Ghalwash, Zach Shahn, et al. G-net: a recurrent network approach to g-computation for counterfactual prediction under a dynamic treatment regime. In *Machine Learning for Health*, pp. 282–299. PMLR, 2021.

Bryan Lim. Forecasting treatment responses over time using recurrent marginal structural networks. *Advances in neural information processing systems*, 31, 2018.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Yuchen Ma, Dennis Frauen, Emil Javurek, and Stefan Feuerriegel. Foundation models for causal inference via prior-data fitted networks. In *International Conference on Learning Representations*, volume 2026, pp. 79065–79098, 2026.

Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Causal transformer for estimating counterfactual outcomes. In *International conference on machine learning*, pp. 15293–15329. PMLR, 2022.

John Miller, Chloe Hsu, Jordan Troutman, Juan Perdomo, Tijana Zrnic, Lydia Liu, Yu Sun, Ludwig Schmidt, and Moritz Hardt. Whynot, 2020. URL https://doi.org/10.5281/zenodo.3875775.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=KSugKcbNf9.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Tom Pollard, Benjamin E. Moody, Li-wei H. Lehman, Brian J. Gow, Chrystinne Fernandes, Chen Xie, Alistair Johnson, Roger G. Mark, and Thomas Heldt. PhysioNet as a global platform for biomedical research. *Nature Health*, 1(8):792–795, August 2026. ISSN 3005-0693. doi: 10.1038/s44360-026-00096-z. URL https://doi.org/10.1038/s44360-026-00096-z.

Andres Potapczynski, Ravi Kiran Selvam, Tatiana Konstantinova, Shankar Ramasubramanian, Malcolm Wolff, Kin G Olivares, Ruijun Ma, Mengfei Cao, Michael W Mahoney, Andrew Gordon Wilson, et al. Time-aware prior fitted networks for zero-shot forecasting with exogenous variables. *arXiv preprint arXiv:2603.15802*, 2026.

Jake Robertson, Arik Reuter, Siyuan Guo, Noah Hollmann, Frank Hutter, and Bernhard Schölkopf. Do-pfn: In-context learning for causal effect estimation. *Advances in Neural Information Processing Systems*, 38:174811–174848, 2026.

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.

James M Robins, Miguel Angel Hernan, and Babette Brumback. Marginal structural models and causal inference in epidemiology. *Epidemiology*, 11(5):550–560, 2000.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Ege Onur Taga, Muhammed Emrullah Ildiz, and Samet Oymak. Timepfn: Effective multivariate time series forecasting with synthetic data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 20761–20769, 2025.

Dennis Thumm and Ying Chen. Interventional time series priors for causal foundation models. *arXiv preprint arXiv:2603.11090*, 2026.

Shirly Wang, Matthew BA McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C Hughes, and Tristan Naumann. Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii. In *Proceedings of the ACM conference on health, inference, and learning*, pp. 222–235, 2020.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=Yd8eHMY1wz.

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.

Amirhossein Zare, Amirhessam Zare, Herlock Rahimi, Reza Salarikia, and Mohammad Kashkooli. Causal longitudinal prior-fitted networks for counterfactual outcome prediction. *arXiv preprint arXiv:2606.05797*, 2026.

A Architecture Details

Throughout this appendix, B denotes the number of tasks per gradient step (Section 4.2), and B_μ the number of tasks per micro-batch, i.e., in one forward pass; n, d_x, d_s, n_a, T, and H retain their definitions from Section 3.2. Because time sequences are zero-indexed, the unified padded temporal dimension is defined with length $T_{\text{ax}} = \max\left(T + H,\, \max_m T_m\right) + 1$. All units within a task share this single time axis, which must be sufficiently long to span both the query's prediction horizon and the maximum duration of any in-context trajectory. Consequently, T_{ax} will exceed $T + H + 1$ whenever a context trajectory satisfies $T_m > T + H$. For any individual unit, temporal positions extending beyond its last time step are padded and masked out as keys during attention. Finally, we write $V = 2 + d_x$ for the width of a task's variate axis (treatment row, outcome, covariates). Tasks in a micro-batch may differ in n, d_x, and T_{ax}, so each axis is zero-padded to its maximum over the micro-batch, and padded units and variates are masked in the same way as padded time steps. Whenever n, T_{ax}, and V describe a batched tensor below, they denote these maxima, e.g., $V = 2 + \max d_x \le 2 + d_x^{\max}$. The initial pre-layer input tensor therefore has shape $(B_\mu,\, n{+}1,\, T_{\text{ax}},\, V,\, d_{\text{model}})$ for every forward pass through the network.

A.1 Tokenization

The scalar values of both outcome and covariate cells are projected into the latent space via a shared linear transformation $\mathbb{R} \to \mathbb{R}^{d_{\text{model}}}$. Notably, the architecture relies on no variate-specific parameters. Instead, the specific identity of each variate is conveyed exclusively through an additive feature embedding. This embedding is indexed over the fixed set $\{0, 1, \ldots, d_x^{\max}{+}1\}$, sized to the largest supported system ($d_x^{\max} = 10$, Table 4), where index 0 designates the treatment channel, index 1 corresponds to the outcome, and indices $2, \ldots, d_x{+}1$ are allocated to a task's d_x covariates. This design choice lets a single parameterization handle tasks of varying dimensionality up to capacity.

Treatments are represented via a one-hot encoding over the treatment space $\mathcal{A}$. The resulting n_a-dimensional indicator vector is linearly mapped to $\mathbb{R}^{d_{\text{model}}}$ and subsequently shifted forward in time by one step, such that the treatment a_{t-1} aligns with the temporal slice t. Because no intervention precedes the initial observation, the zeroth slice is populated with category $0 \in \mathcal{A}$ prior to encoding. Although this maps the initial step to a valid treatment level rather than a distinct missing token, we operate under the convention that level 0 corresponds to a reference level.

Furthermore, static covariates $\boldsymbol{s}$ are processed through a two-layer MLP ($\mathbb{R}^{d_s} \to \mathbb{R}^{d_{\text{model}}} \to \mathbb{R}^{d_{\text{model}}}$) equipped with GELU activations. The resulting static representation is broadcast across the temporal and variate dimensions of its respective unit, functioning as a uniform per-unit bias applied to every token. This approach avoids cluttering the attention mechanism with an arbitrary sequence of static tokens and keeps the model applicable when statics are absent, in which case $\boldsymbol{s}$ is set to $\mathbf{0} \in \mathbb{R}^{d_s}$.

The additive embeddings (comprising feature, unit-type, and static terms) are applied uniformly to all tokens across every unit. Two distinct masking mechanisms then operate on this grid. The first hides unobserved content while keeping the token in place; the second removes a token from the attention computation altogether. The first applies to the query's prediction horizon. The unobserved scalars at $t > T$ are zeroed at the input, and their scalar-projection term is replaced by a single learned mask token, shared across all variates and horizon time steps. The additive terms are left untouched, so the cell retains its variate identity through the feature embedding. Its temporal coordinate is likewise preserved: RoPE is injected inside the temporal attention block, applied to queries and keys after projection, so position is supplied by the attention operation itself and does not depend on the content of the embedding. A masked horizon cell therefore sits at a known variate and a known time step while carrying no information about the value to be predicted. The second applies to every unit. Positions beyond a unit's last time step, T_m for an in-context unit and $T+H$ for the query, are zero-padded and excluded as keys, so they contribute nothing to the output.

A.2 Attention axes and cost

Each attention reshapes the grid so that the attended axis is the sequence axis and the remaining axes are folded into the batch:

- Temporal: $\big(B_\mu(n{+}1)V,\ T_{\text{ax}},\ d_{\text{model}}\big)$.
- Variate: $\big(B_\mu(n{+}1)T_{\text{ax}},\ V,\ d_{\text{model}}\big)$.
- Context: $\big(B_\mu T_{\text{ax}}V,\ n{+}1,\ d_{\text{model}}\big)$.

By factoring out the total token count $B_{\text{eff}} = B_\mu(n{+}1)T_{\text{ax}}V$ of a micro-batch, the computational complexity of a single layer evaluates to $O\big(B_{\text{eff}}\, d_{\text{model}}(T_{\text{ax}} + V + n + 1)\big)$ for the attention operations and $O\big(B_{\text{eff}}\, d_{\text{model}}^2\big)$ for the linear projections and feed-forward networks. Crucially, this demonstrates that the computational cost grows linearly with the total number of tokens and scales with the sum of the three dimension lengths, circumventing the prohibitively expensive quadratic scaling (driven by the product of these lengths) that would result from flattening the grid into a single joint attention operation. For efficient execution, the attention mechanisms across all three axes are implemented using PyTorch's (Paszke et al., 2019) fused scaled dot-product attention. Furthermore, RoPE is applied exclusively along the temporal dimension. No positional signal is injected into the other two dimensions: the variate axis carries no positional encoding, and variate identity comes only from the index-based feature embedding. The unit axis strictly omits positional encodings to guarantee that the model remains permutation-equivariant with respect to the context.

A.3 Mask Details

Temporal. The temporal mask is what makes MTS-CFM a direct multi-step predictor: the query's horizon states are targets rather than keys on this axis, so the entire horizon is produced in one forward pass from the observed history and the treatment plan, with no autoregressive feedback. During temporal attention, in-context units process their entire trajectories bidirectionally, as they function strictly as encoders and do not contribute to the training loss. Likewise, the query reads its observed history $0{:}T$ bidirectionally in every row. Inside the horizon, its target cells read that history alone and never another horizon token, including those at earlier horizon steps, while the

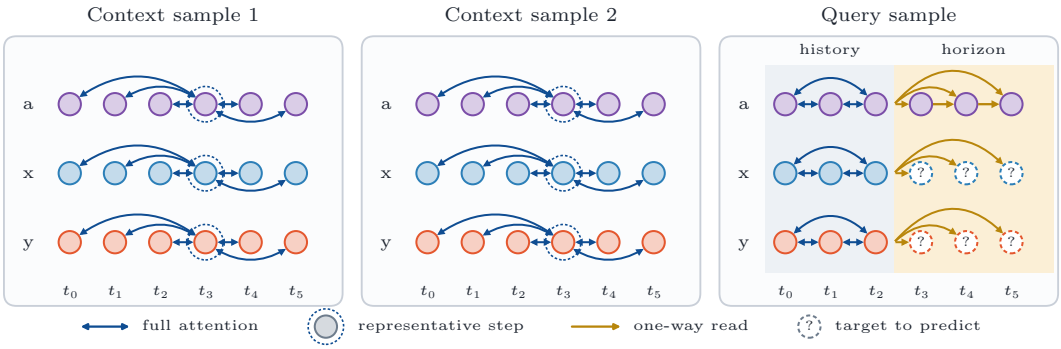

Figure 3: Temporal attention. Context units read their trajectories bidirectionally. In the horizon, the target cells read only the history. The treatment row, the only one observed there, is read causally.

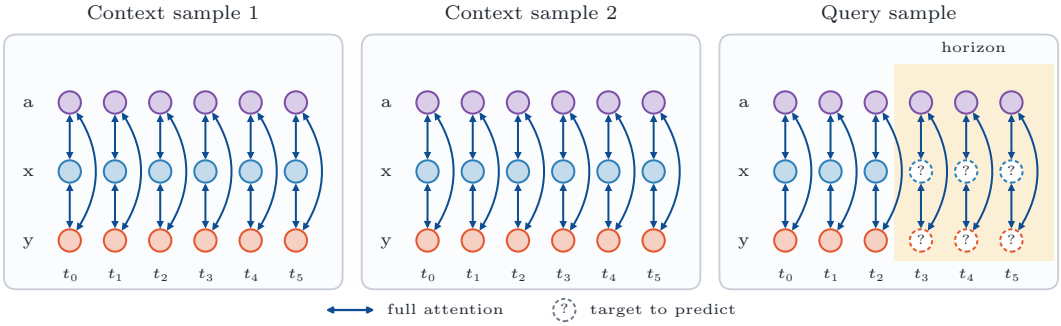

Figure 4: Variate attention. All channels attend freely within a time slice, so the horizon's mask tokens read the planned treatment. This is the route by which interventions enter the forecast.

treatment row, the only row observed there, reads causally along the plan (Figure 3). Because in-context tokens are able to aggregate information from their complete sequences, the subsequent same-t context attention mechanism can supply the query with future evidence derived from the cohort. Crucially, this transfers valuable counterfactual signals without ever violating the temporal causal cutoff of the query's own pathway.

Furthermore, because temporal attention operates independently within each variate channel, a masked target position in the horizon never directly reads the treatment row. Instead, the treatment row carries the plan forward on its own: since slice t encodes $\bar{a}_{t-1}$, the causal mask lets the treatment token at slice $T+h$ read the observed treatment history and the planned treatments $\bar{a}_T, \ldots, \bar{a}_{T+h-1}$, but no later one, and variate attention at the same slice then injects this summary into the query cells. This enforces correct physical dependencies by isolating the predicted state to rely exclusively on the sequence of treatments logically upstream of it. These dual masking regimes are explicitly visible within the treatment row (Figure 3, right panel), demonstrating bidirectional reading over the observed history side-by-side with causal reading across the prediction horizon.

Variate. Along the variate dimension, no structural masking is imposed. Within any given time slice, all covariates, outcome and treatment freely attend to one another (Figure 4). The sole exceptions are the inactive, padded variate positions that exceed the task-specific dimensionality d_x. This fully connected structure applies equally to the query's prediction horizon: the learned mask tokens representing the unobserved outcome and covariates attend both to each other and to the provided future treatment. Consequently, this attention mechanism serves as the primary pathway through which planned interventions are integrated into the predicted future states.

Context. This dimension adopts the context–query separation introduced by the PFN architecture (Müller et al., 2022), though in our design it is applied at the granularity of individual (time, variate) cells rather than globally per unit. Let i denote the query row and j the key column along the

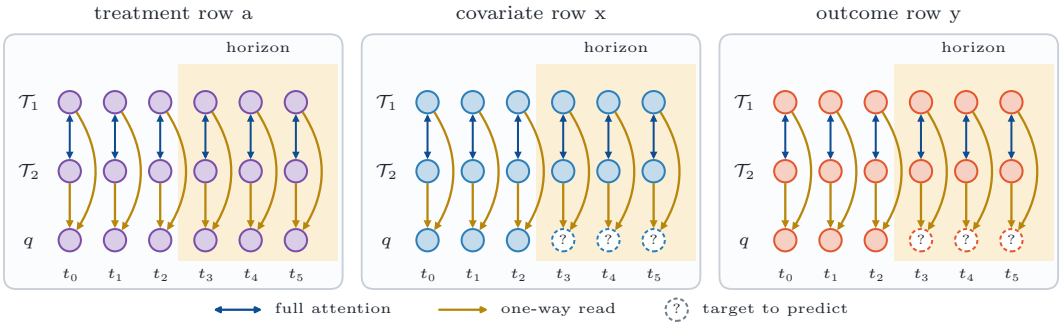

Figure 5: Context attention. The cohort is fully connected internally; the query reads it but never the reverse, keeping the prediction permutation-invariant and the cohort cacheable.

unit axis, where the first n columns correspond to the in-context trajectories and the final column represents the target query. The structural attention mask is defined as $\mathbf{1}[j < n] \vee \mathbf{1}[i = j]$. To handle missing data and variable sequence lengths, this logical mask is intersected with the per-cell observation mask. The self key $i = j$ is exempt from this intersection and always retained, so every cell has at least one valid key, including a query horizon cell at a time slice where no in-context unit is observed. Ultimately, this allows the in-context cohort to be fully connected internally, while the query attends to the cohort in a strictly unidirectional manner (Figure 5). This one-way information flow operates at every time index and across all variate rows, extending into the prediction horizon.

Within the context cohort, information is exchanged bidirectionally, yielding increasingly abstracted representations at greater network depths. Hence, an in-context token assimilates its unit's complete temporal trajectory alongside the broader population's dynamics at that specific coordinate. Consequently, in deeper layers, the query attends to a highly refined, high-level synthesis of the underlying process rather than raw input features. Furthermore, because the strictly unidirectional attention from the query to the cohort is maintained across every layer, the cohort's internal representation remains exclusively a function of $\mathcal{D}_n$. No information from the query is ever allowed to perturb the context embedding. This strict architectural isolation guarantees that the predictive distribution $q_{\boldsymbol{\theta}}$ is permutation-invariant with respect to the context set, ensuring the output perfectly matches that of a single-query forward pass. Computationally, this design also allows the cohort's representations to be evaluated once and efficiently cached for reuse across multiple queries.

A.4 Prediction Head and Loss

The head is evaluated only on the gathered query horizon tokens, of shape $(B_\mu,\, H,\, V{-}1)$, where padded covariate channels are excluded from the loss. Throughout this section, c denotes a valid horizon cell, that is, one (step, channel) pair of the query's outcome or covariates at a time $T{+}h$, and z_c denotes its target value.

Mixture density. For each cell, the head maps the token to the parameters of a K-component univariate Gaussian mixture,

$$q_{\boldsymbol{\theta}}(z_c) = \sum_{k=1}^{K} \pi_{c,k}\, \mathcal{N}\big(z_c \mid \mu_{c,k},\, \sigma_{c,k}^2\big), \tag{3}$$

where the weights $\pi_{c,k}$ are obtained by a softmax over the mixture logits, the component means $\mu_{c,k}$ are bounded by a scaled $\tanh$, and the scales $\sigma_{c,k}$ are constrained to $[\sigma_{\min}, \sigma_{\max}]$ (Table 4). The predictive mean and variance of the cell follow in closed form,

$$\hat{z}_c = \sum_k \pi_{c,k}\, \mu_{c,k}, \qquad \hat{v}_c = \sum_k \pi_{c,k}\big(\sigma_{c,k}^2 + \mu_{c,k}^2\big) - \hat{z}_c^{\,2}. \tag{4}$$

Point predictions used in evaluation are the mean $\hat{z}_c$.

NLL. The primary term is the NLL of the target under the mixture,

$$\ell_{\text{NLL}}(c) = -\log q_{\boldsymbol{\theta}}(z_c). \tag{5}$$

As a strictly proper scoring rule, it trains the full predictive distribution of each cell, so that $q_{\boldsymbol{\theta}}$ approximates the per-variable, per-step marginal of the CID (Section 4.1).

Huber mean. The Huber loss is introduced because NLL alone can be reduced by widening the predictive distribution rather than improving its mean, which can degrade the point estimates used during evaluation. We therefore add a robust regression loss on the mixture mean. With residual $r_c = \hat{z}_c - z_c$,

$$\ell_{\text{mean}}(c) = \begin{cases} \frac{1}{2}\, r_c^2, & |r_c| \le \delta, \\ \delta\big(|r_c| - \frac{1}{2}\delta\big), & |r_c| > \delta. \end{cases} \tag{6}$$

This loss is quadratic for small residuals and linear for large ones, so its gradient stays bounded for large residuals. It is preferred over the squared error for this reason, as it stabilizes training.

Concentration. To prevent the mixture from collapsing into a single Gaussian mode, we penalize the largest mixture weight when it exceeds a threshold τ,

$$\ell_{\text{conc}}(c) = \max\Big(0,\ \max_k \pi_{c,k} - \tau\Big). \tag{7}$$

The term is zero whenever no component holds more than a fraction τ of the mass, so it leaves unimodal predictions possible but stops a single component from absorbing all (Zare et al., 2026).

Cell weights. Each cell receives a weight w_c, with $w_c = 1$ on the covariate channels and $w_c = w_{\text{y}} \ge 1$ on the outcome channel. When $w_{\text{y}} = 1$, the outcome and covariates contribute equally; when $w_{\text{y}} > 1$, the outcome contributes more to the loss, and the model prioritizes predicting it accurately. We set $w_{\text{y}} = d_x$ (Table 5), so at each horizon step the outcome carries as much total weight as all covariates combined, independently of the task's dimensionality.

Objective. Combining the three per-cell terms with weights λ_{mean} and λ_{conc}, and normalizing by the total cell weight, gives

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{\sum_c w_c} \sum_c w_c \Big[\ell_{\text{NLL}}(c) + \lambda_{\text{mean}}\, \ell_{\text{mean}}(c) + \lambda_{\text{conc}}\, \ell_{\text{conc}}(c)\Big]. \tag{8}$$

The normalizer $\sum_c w_c$ is the weighted number of supervised cells of a micro-batch, which is also the weight with which each micro-batch enters the accumulated gradient (Section 4.2).

A.5 Memory

Because the tensor grid is instantiated across five dimensions and each layer executes three distinct attention mechanisms, activation memory, emerges as the primary hardware constraint. To mitigate this, we apply selective activation checkpointing (Chen et al., 2016; Korthikanti et al., 2023) across every layer within the network. Crucially, this selection is performed at the granularity of individual operations rather than entire layers. Specifically, we preserve the intermediate outputs of computationally intensive operations, namely matrix multiplications and fused scaled dot-product attention kernels. Conversely, all lightweight operations, including normalizations, non-linear activations, tensor reshapes, masking, and residual additions, are discarded from memory and subsequently recomputed during the backward pass. This strategic retention optimally balances the memory-compute trade-off: by discarding only those states that are inexpensive to regenerate, we incur a minimal computational overhead. This efficiency strictly liberates sufficient memory to accommodate larger cohort sizes n and sequence lengths T_{ax} under a fixed hardware budget.

A.6 Hyperparameters

Tables 4 and 5 list the hyperparameters of the single pretrained model used for all experiments. Table 4 covers the architecture: the backbone, the task dimensions that set the size of the token grid,

and the parameterization of the mixture head (Appendix A.4). Table 5 gives the weights of the loss terms in Equation (8) together with the optimization schedule. The same model is applied to every benchmark without fine-tuning.

The resulting model contains 8 million trainable parameters. We have utilized a single A100 GPU with 80 GB of memory for training.

Table 4: Architecture hyperparameters.

Symbol	Description	Value
Backbone		
d_{model}	Token embedding width	256
d_{ff}	Feed-forward hidden width	1024
L	Layers (each: three attentions + FFN)	6
n_{heads}	Attention heads, all three axes	8
	FFN activation	GELU
	Normalisation	pre-norm LayerNorm
	RoPE base frequency (temporal axis only)	10^4
Task dimensions		
$T_{\max}$	Maximum time-axis length T_{ax} (RoPE cache)	66
H	Forecast horizon	5
$d_x^{\max}$	Maximum number of covariates	10
n_a	Treatment categories $\lvert\mathcal{A}\rvert$ (one-hot)	4
d_s	Static covariate dimension	5
n	In-context units per task (training)	$[3, 200]$
$n_{\max}$	Maximum in-context units per task (evaluation)	500
Predictive head		
K	Mixture components per cell	5
	Mixture-logit temperature	1.0
	Mean bound ($\tanh$ scale)	7.0
$\sigma_{\min}$	Scale floor	0.02
$\sigma_{\max}$	Scale ceiling	2.0

Table 5: Objective and optimization hyperparameters.

Symbol	Description	Value
Objective		
λ_{mean}	Weight on the Huber mean term	1.0
δ	Huber transition point	3.0
τ	Concentration hinge threshold on $\max_k \pi_k$	0.9
λ_{conc}	Weight on the concentration hinge	0.1
w_{y}	Outcome channel weight	d_x
Optimization		
	Optimizer	AdamW
	Peak learning rate	0.005
	Warmup	Linear
	Schedule	Cosine
	Weight decay	0.01
	Gradient clipping	10
B	Tasks per gradient step	256
B_μ	Tasks per micro-batch	4
	Gradient accumulation steps	64
	Training steps	10,000
	Warmup steps	500

B PRE-TRAINING PRIOR

In this section, we outline the data-generating prior based on Zare et al. (2026), the procedure for assembling a single training episode, and the optimization process. Our mathematical notation aligns with Section 3.2: d_x represents a task's total number of covariates, $\mathbf{x}_t \in \mathbb{R}^{d_x}$ denotes the covariate state, y_t is the outcome, and $\mathrm{a}_t \in \mathcal{A}$ indicates the applied treatment. Additionally, $\mathbf{s}$ stands for the static covariates, while H defines the time horizon. Within a single unit, we drop the sample index, and the i-th variate of $\mathbf{x}_t$ is denoted by $\mathrm{x}_{i,t}$, and we use $b(\mathrm{a}) = (\mathrm{a} \bmod 2, \lfloor \mathrm{a}/2 \rfloor)$ to express the two-bit binary encoding of a treatment.

B.1 SAMPLING

A single task corresponds to a sampled temporal SCM, denoted as $\mathbb{P} \sim \Pi$, which is generated through the following process.

Graph. We first sample the system width $d_x \sim \mathcal{U}\{1, \ldots, 10\}$ and the maximum lag $K \sim \mathcal{U}\{1, 2\}$. To guarantee an acyclic structure along the variate axes, the instantaneous (within-slice) adjacency matrix $\boldsymbol{G}^{(0)}$ is formulated as a strictly lower-triangular Erdős–Rényi graph. Its edge probability is defined as $p = 0.1 + 0.5\beta$, where $\beta \sim \mathrm{Beta}(2, 2)$. For the temporal dependencies, the lagged adjacency matrices $\boldsymbol{G}^{(k)}$ (for $k = 1, \ldots, K$) are modeled as dense Bernoulli matrices. Their edge probabilities decay with distance as $p\,\gamma_{\text{lag}}^{k}$, where $\gamma_{\text{lag}} \sim \mathcal{U}(0.4, 0.8)$, ensuring that older states exert less influence. Finally, edge weights are drawn from zero-mean Gaussian distributions on this sampled support: $W_{ij}^{(0)} \sim \tilde{W}_{ij}^{(0)}\, G_{ij}^{(0)}$ for contemporaneous edges and $W_{ij}^{(k)} \sim \tilde{W}_{ij}^{(k)}\, G_{ij}^{(k)}$ for lagged edges, where $\tilde{W}_{ij}^{(0)} \sim \mathcal{N}(0, \sigma_w^2)$ and $\tilde{W}_{ij}^{(k)} \sim \mathcal{N}(0, (0.7\sigma_w)^2)$ with the base standard deviation drawn as $\sigma_w \sim \mathcal{U}(0.3, 1.0)$.

Generic mechanism. A variate not claimed by a motif evolves as a sparse nonlinear autoregression,

$$\mathrm{x}_{i,t} \;=\; \alpha_i\, \mathrm{x}_{i,t-1} \;+\; \phi_i\Big(\sum_{j<i} W_{ij}^{(0)}\, \mathrm{x}_{j,t} \;+\; \sum_{k=1}^{K} \sum_{j} W_{ij}^{(k)}\, \mathrm{x}_{j,t-k} \Big) \;+\; \epsilon_{i,t}, \tag{9}$$

where the within-slice summation incorporates the current slice as it is being constructed, whereas the lagged and persistence terms draw from a buffer that is delayed by one time step. The nonlinear function ϕ_i is selected uniformly from $\{\mathrm{id}, \tanh, \sin, \cos, |\cdot|, (\cdot)^2, \mathrm{ReLU}, \mathrm{softplus}\}$. The persistence parameter is either set to $\alpha_i = 0$ with a probability of 0.5, or sampled from $\alpha_i \sim \mathcal{U}(0.5, 1.0)$. For the noise, each variate independently selects a distribution family from Gaussian, uniform, Laplace. The scale of this noise is 0 half the time, meaning roughly 50% of the covariates are deterministic given their parents, and otherwise it is computed as the task-level baseline σ_0 scaled by a factor sampled from $\mathcal{U}(0.5, 1.25)$. The task-level baseline scale σ_0 is drawn from $\mathcal{U}(0, 0.05)$ with probability 0.6, and from $\mathcal{U}(0.05, 0.20)$ otherwise. To ensure the recursion remains bounded, states are clipped into the range $[-10, 10]$ during generation. Finally, to introduce a minority of non-stationary tasks, there is a 0.12 probability that a task samples a second mechanism on the identical graph support. This new mechanism, complete with fresh weights and nonlinearities, takes effect at a random switch point located in the first two-thirds of the sequence.

Dynamical motifs. Rather than using the generic mechanism, some covariates chosen via random permutation might be controlled by one of five structured channels (Table 6). These motifs serve two main purposes. First, they embed qualitative behaviors typical of real-world systems (such as accumulation with decay, bounded saturation, set-point regulation, closed-loop control, and slow proxy measurements) into the prior. Second, they provide the sole pathway for the treatment to influence the covariate state. The generic mechanism lacks a treatment term, whereas four of the five motifs respond to treatment directly, through the current treatment or the decaying treatment memory; the readout motif carries treatment effects only indirectly, by tracking another covariate.

Heterogeneity, statics and the behaviour policy. Each unit, whether a query or a cohort member, samples a single latent factor $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_3)$. This factor influences both the initial state and the

Table 6: Dynamical motifs. Each claims a distinct covariate i with the listed probability and replaces the generic mechanism there. $\mathbf{m}_t$ is the decaying two-bit treatment memory of the behaviour policy; $\mathrm{l}_t \geq 0$ is a signal derived from it by multiplying a random factor drawn from $\mathcal{N}(0, 0.45^2)$, $j(i)$ a source covariate, and $\epsilon_{i,t}$ the covariate's own noise, omitted from the column for brevity. The saturating channel's baseline flow additionally carries a bounded modulation by the feedback covariate when the task has one.

Motif	p	Update for $\mathrm{x}_{i,t}$ (plus $\epsilon_{i,t}$)	Ranges
Action memory	0.25	$\delta_i \mathrm{x}_{i,t-1} + \boldsymbol{w}_i^\top b(\mathrm{a}_{t-1}) + \boldsymbol{v}_i^\top \mathbf{m}_t$	$\delta_i \in [0.72, 0.97]$
Saturating	0.25	$\mathrm{x}^+_{i,t-1} + r_i b_i \big(1 - g_i \frac{\mathrm{l}_t}{h_i + \mathrm{l}_t}\big) - r_i \mathrm{x}^+_{i,t-1}$, clipped to $[0, 6]$	$r_i \in [0.02, 0.15]$, $b_i \in [0.5, 1.5]$, $g_i \in [0.25, 0.95]$, $h_i \in [0.3, 2.0]$
Homeostatic	0.25	$\mathrm{x}_{i,t-1} + \kappa_i(\mu_i - \mathrm{x}_{i,t-1}) + \boldsymbol{w}_i^\top b(\mathrm{a}_{t-1})$	$\kappa_i \in [0.03, 0.15]$, $\mu_i \in [-0.5, 0.5]$
Feedback	0.25	$\rho_i \mathrm{x}_{i,t-1} + \nu_i(\eta_i - \mathrm{x}_{j(i),t-1}) + \boldsymbol{w}_i^\top b(\mathrm{a}_{t-1})$	$\rho_i \in [0.65, 0.95]$, $\nu_i \in [0.10, 0.90]$
Readout	0.20	$\varrho_i \mathrm{x}_{i,t-1} + (1 - \varrho_i)\, \mathrm{x}_{j(i),t}$	$\varrho_i \in [0.70, 0.97]$

policy weights, thereby linking treatment propensity with baseline severity for each individual. The observed static covariates $\mathbf{s} \in \mathbb{R}^5$ act as a noisy proxy for $\mathbf{z}$, comprising one continuous variable, a complementary binary pair, and a two-of-three categorical indicator. To ensure the model does not always rely on baseline features, these statics are active in only 30% of the tasks and are otherwise set to zero. The dependence of the initial state and policy weights on $\mathbf{z}$ is expressed as

$$\mathbf{x}_0 \;\approx\; \boldsymbol{U}_x\, \mathbf{z} + \boldsymbol{\epsilon}_0, \qquad \boldsymbol{w}_u \;=\; \kappa\, \boldsymbol{w}_u^{\mathrm{pop}} + \boldsymbol{U}_u \mathbf{z}, \qquad u \in \{0, 1\}, \tag{10}$$

where $\boldsymbol{\epsilon}_0 \sim \mathcal{N}(\mathbf{0}, 0.1^2 \boldsymbol{I})$ and $\boldsymbol{w}_u^{\mathrm{pop}} \sim \mathcal{N}(\mathbf{0}, \sigma_\pi^2 \boldsymbol{I})$ with $\sigma_\pi \sim \mathcal{U}(0.5, 2.0)$ drawn separately for each bit. The entries of $\boldsymbol{U}_x$ and $\boldsymbol{U}_u$ are i.i.d. $\mathcal{N}(0, \sigma_x^2)$ and $\mathcal{N}(0, \sigma_U^2)$, with $\sigma_x, \sigma_U \sim \mathcal{U}(0.1, 1.0)$ and σ_U shared across both bits. The approximation in $\mathbf{x}_0$ reflects that coordinates belonging to dynamical motifs are instead initialised near their motif-specific baselines.

Treatment assignments are generated bit by bit using a logistic policy that incorporates the current state, a decaying memory of prior treatments, and the latent factor $\mathbf{z}$ via $\boldsymbol{w}_u$:

$$b_{u,t} \sim \mathrm{Bernoulli}\big(\sigma(\boldsymbol{w}_u^\top \mathbf{x}_t + c_u + \psi_u\, \mathrm{m}_{u,t})\big), \qquad \mathrm{m}_{u,t} = \lambda_u\, \mathrm{m}_{u,t-1} + b_{u,t-1}, \tag{11}$$

where $\mathbf{m}_t = (\mathrm{m}_{0,t}, \mathrm{m}_{1,t})$ is the two-bit treatment memory that also drives the motifs in Table 6. The task-specific intercept $c_u \sim \mathcal{N}(0, 1)$ establishes baseline prevalence. The memory decay $\lambda_u \sim \mathcal{U}(0.5, 0.95)$ and the strictly negative weight $\psi_u \sim \mathcal{U}(-2.0, -0.5)$ ensure that recent treatments reduce the likelihood of subsequent exposure. The applied treatment is constructed as $\mathrm{a}_t = b_{0,t} + 2b_{1,t} \in \mathcal{A}$. While this bit-wise approach yields heterogeneous policies, the model only observes the resulting four-level treatment. The policy strength κ scales only the population component of the state weights in equation 10. It equals 0 with probability 0.08, 1 with probability 0.20, and otherwise takes an integer value from $\{2, \ldots, 5\}$. Because both treatment assignment and the baseline state hinge on $\mathbf{z}$, and assignments react to states modified by previous treatments, the generated tasks exhibit varying degrees of time-dependent confounding. The explicit probability mass at $\kappa = 0$ removes the population-level dependence of assignment on the current state, leaving only the weaker unit-specific term $\boldsymbol{U}_u \mathbf{z}$, and thereby maintains the necessary overlap.

Outcome. The outcome monitors the state via R_t. With probability 0.4, R_t is a single state variate, chosen with preference for coordinates that belong to dynamical motifs. Otherwise, R_t is an unobserved affine projection $\boldsymbol{w}_R^\top \mathbf{x}_t + b_R$ with $b_R \sim \mathcal{N}(0, 1)$, where $\boldsymbol{w}_R \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ with probability 0.5 and $\boldsymbol{w}_R = \mathbf{0}$ otherwise. In the latter case the outcome carries no state signal. In one-quarter of the projection tasks, R_t is instead a convex combination $\alpha R_t^{(1)} + (1 - \alpha) R_t^{(2)}$ of two such projections, with $\alpha \sim \mathcal{U}(0.35, 0.65)$ and the second projection's weights always drawn from $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Starting from $\mathrm{y}_0 = R_0 + \eta_0$, the scalar outcome evolves autoregressively according to

$$\mathrm{y}_t \;=\; \rho\, \mathrm{y}_{t-1} \;+\; (1 - \rho)\big(g\, R_t + \boldsymbol{w}_a^\top b(\mathrm{a}_{t-1}) + \boldsymbol{w}_c^\top \mathbf{m}_t + \tau\, t\big) \;+\; \eta_t, \tag{12}$$

where $\mathbf{m}_t$ is the treatment memory from equation 11. The coefficients are $\rho \sim \mathcal{U}(0.35, 0.90)$, $g \sim \mathcal{U}(0.35, 1.20)$, $\boldsymbol{w}_a \sim \mathcal{N}(\mathbf{0}, \sigma_a^2 \boldsymbol{I}_2)$ with $\sigma_a = 0.25$ (increased to $0.25\sqrt{2}$ when the task contains a dynamical motif), $\boldsymbol{w}_c \sim \mathcal{N}(\mathbf{0}, 0.1^2 \boldsymbol{I}_2)$, and $\tau \sim \mathcal{N}(0, 0.015^2)$. The noise η_t is Gaussian with a scale drawn from $\mathcal{U}(0, 0.06)$ with probability 0.7 and from $\mathcal{U}(0.06, 0.16)$ otherwise.

B.2 IDENTIFIABILITY

Well-defined interventions and consistency. The treatment is categorical with four levels, and a plan sets those levels directly. Hence, an intervention leaves no ambiguity about what was done. Factual trajectories come from forward simulation of the sampled temporal SCM, and interventionals from replacing the policy with $\bar{\boldsymbol{a}}$ while reusing the same exogenous draws. The observed trajectory is therefore a potential trajectory at the treatment sequence that was actually applied.

No interference. Each unit draws its own latent factor $\mathbf{z}$, initial state and noise. Mechanism parameters are shared within a task, but no unit's variables enter another unit's equations. Thus, one unit's treatment cannot change another's trajectory.

Sequential ignorability. By Equation (11), a_t is a function of the current state, the treatment memory, the statics, the unit's policy weights, and an independent coin flip. Given $\mathcal{H}_t$, the trajectory under a plan depends only on the last few (lagged) states, the task's mechanism parameters, the plan itself, and an independent noise draw. Hence, $(\mathbf{X}, \mathbf{y})_{t+1:T+H}(\bar{\boldsymbol{a}}) \perp \mathrm{a}_t \mid \mathcal{H}_t, \boldsymbol{s}$ at every step.

Positivity. The policy is logistic, so each treatment bit has probability strictly between 0 and 1 and every treatment sequence can occur. The prior also spans a range of overlap, from weakly state-dependent assignment to sharply state-dependent assignment.

C BENCHMARKS

We use the same four evaluation domains, data-generating code, and evaluation protocol of Causal-LongPFN (Zare et al., 2026) as provided in the their repository (`https://github.com/Amirhossein-Zare/causal-long-pfn`) for our model, so our results are directly comparable to the baselines they report. Below, we briefly describe each domain.

Common protocol. Every domain is cast into the interface of Section 3.2: at each discrete step t a unit has a time-varying state $\mathbf{x}_t$, a scalar outcome y_t, a treatment $\mathrm{a}_t \in \mathcal{A} = \{0, 1, 2, 3\}$, and optionally static covariates $\mathbf{s}$. Raw trajectories have length 60, forecasts start only after at least 10 observed steps, and the horizon is $H = 5$. A task is defined by a confounding level γ (ten levels), a context size $n \in \{40, 80, 160, 320, 500\}$ and a random seed. In the simulated domains, γ scales how strongly the behaviour policy reacts to the current patient state, and hence the strength of time-varying confounding. In MIMIC-III, confounding cannot be varied, so its tasks instead differ in which patients are randomly drawn as context and query. Outcomes are normalized at scoring time using statistics of the context set only.

Interventional and observational queries. Cancer, warfarin and HIV are branchable simulators: from a factual patient state we replay the same patient-specific dynamics under alternative treatments, which yields ground-truth interventional targets. One-step queries evaluate all four treatments at the branching point, and horizon-5 queries apply a randomly sampled treatment sequence of length H. MIMIC-III contains only observed ICU care, so both one-step and horizon-5 queries are factual predictions under the treatments that were actually administered. MIMIC-III results should therefore be read as temporal prediction results rather than as validation of interventional effects. Table 7 summarizes the four domains.

Table 7: Evaluation domains, following Zare et al. (2026). The first three provide interventional labels by simulation; MIMIC-III is observational only.

Domain	d_x	Treatments ($\mathrm{a}_t = 0, 1, 2, 3$)	Outcome y_t	Labels
Cancer	0	none, chemo, radio, chemo+radio	$\log(1 + \text{clipped tumor volume})$	interventional
Warfarin	9	0, 2, 5, 10 mg/day	INR	interventional
HIV	5	none, PI, RTI, RTI+PI	$\log_{10}(1 + \text{free virus})$	interventional
MIMIC-III	9	none, vaso, vent, vaso+vent	diastolic blood pressure	observational

C.1 CANCER TUMOR GROWTH

This is the non-small-cell lung cancer simulator of Geng et al. (2017), the standard benchmark for longitudinal interventional prediction (Lim, 2018; Bica et al., 2020; Melnychuk et al., 2022; Li et al., 2021). Tumor volume grows toward a carrying capacity and is reduced by chemotherapy and radiotherapy, both of which are assigned based on recent tumor size.

Dynamics. The treatment $\mathrm{a}_t \in \{0, 1, 2, 3\}$ splits into a chemotherapy indicator c_t and a radiotherapy indicator r_t via $(c_t, r_t) = b(\mathrm{a}_t)$. Let V_t denote the raw tumor volume, and let $\mathrm{Vol}(D) = \pi D^3/6$ convert a diameter D into a volume. Chemotherapy builds up a drug concentration that halves every step, $C_t = \frac{1}{2}C_{t-1} + 5c_t$, and radiotherapy gives an immediate dose $R_t = 2r_t$. For a single patient the volume evolves as

$$V_{t+1} = V_t\Big(1 + \rho \log \tfrac{\mathrm{Vol}(30)}{V_t} - \beta_c C_t - \alpha R_t - \beta_r R_t^2 + \epsilon_t\Big), \qquad \epsilon_t \sim \mathcal{N}(0, 10^{-4}), \tag{13}$$

where $\mathrm{Vol}(30)$ is the carrying capacity. The logarithmic term is Gompertz growth, the C_t term is the chemotherapy effect, and the two R_t terms are the radiotherapy effect. The growth rate ρ, chemosensitivity β_c, radiosensitivity α and noise ϵ_t are specific to each patient, with $\beta_r = \alpha/10$.

Behaviour policy. Let $\bar{D}_t$ be the average tumor diameter over the last 15 steps. Chemotherapy and radiotherapy are each given independently with probability

$$p_t = \sigma\Big(\frac{\gamma}{D_{\max}}\Big(\bar{D}_t - \frac{D_{\max}}{2}\Big)\Big), \tag{14}$$

where σ is the sigmoid function and $D_{\max} = 13$. Larger tumors are therefore treated more often, and γ controls how strongly, which induces time-varying confounding.

Outcome. The outcome is the log of the tumor volume, clipped at threshold $\mathrm{Vol}(D_{\max})$:

$$\mathrm{y}_t = \log\big(1 + \min\{V_t, \mathrm{Vol}(D_{\max})\}\big). \tag{15}$$

Queries. A query asks for a patient's outcome under one treatment plan at one point in their timeline. Queries are generated by sliding that prediction point forward one step at a time, beginning once $T_{\min} = 10$ steps of history have been observed and ending near the end of the $T_{\max} = 60$ step timeline, which gives 49 prediction points for one-step queries and 50 for horizon queries. At each point the simulator supplies the ground truth: under all $n_a = 4$ treatments for a one-step query, and under $2H = 10$ plans drawn at random from $\mathcal{A}^H$ for a horizon query, as the $n_a^H = 1024$ possible plans are too many to enumerate. A task therefore contributes $n_a \times 49 = 196$ one-step and $2H \times 50 = 500$ horizon-5 queries, fewer when a patient dies or recovers early.

C.2 WARFARIN PK/PD

This is a semi-mechanistic simulator of warfarin dosing, built on standard pharmacokinetic/pharmacodynamic (PK/PD) models of the drug (Holford, 1986; Consortium, 2009; Hamberg et al., 2010). Each step covers four hours. The drug is absorbed, cleared, and reaches an effect site, where it slows the production of vitamin-K-dependent clotting factors. The resulting drop in clotting factors raises the INR, which is the outcome.

Treatment. The treatment $\mathrm{a}_t \in \{0, 1, 2, 3\}$ selects a daily dose of 0, 2, 5 or 10 mg. As in all domains, a_t acts on the transition from step t to step $t{+}1$.

Pharmacokinetics. The drug passes through three stages: the gut, the plasma, and the effect site. Let G be the amount of drug in the gut, C_p the plasma concentration, and C_e the effect-site concentration. Within each step, they follow

$$\dot{G} = -k_a G, \qquad \dot{C}_p = \frac{k_a G}{V_d} - \frac{\mathrm{CL}}{V_d} C_p, \qquad \dot{C}_e = k_{e0}\,(C_p - C_e), \tag{16}$$

where k_a is the absorption rate, V_d the volume of distribution, CL the clearance, and k_{e0} the rate at which the effect site catches up with the plasma. The chosen dose enters the gut stage.

Pharmacodynamics. The effect-site concentration inhibits clotting-factor synthesis through a sigmoid $E_{\max}$ curve,

$$I(t) = E_{\max} \frac{C_e(t)^\lambda}{C_e(t)^\lambda + \mathrm{EC}_{50}^\lambda}, \tag{17}$$

with Hill coefficient λ. Each clotting factor $F \in \{\mathrm{II}, \mathrm{VII}, \mathrm{X}, \mathrm{PC}\}$ is produced at a rate set by the vitamin-K level VK and reduced by the inhibition, and it is removed at its own turnover rate k_F:

$$\dot{F}(t) = k_F\big(\mathrm{VK}(t) \cdot (1 - s_F I(t)) - F(t)\big), \tag{18}$$

where s_F is the factor's sensitivity to the drug. Because factors turn over slowly, the anticoagulant effect lags the dose.

INR. The INR is read out from how far the factors fall below their normal level of 1.

$$\Delta(t) = 1.30\,[1 - F^{\mathrm{VII}}(t)]_+ + 0.95\,[1 - F^{\mathrm{X}}(t)]_+ + 0.80\,[1 - F^{\mathrm{II}}(t)]_+. \tag{19}$$

The deficit is mapped to a prothrombin-time ratio and then to the INR:

$$\mathrm{PT}(t) = 1 + 1.6\,\Delta(t) + 0.70\,\Delta(t)^2, \qquad \mathrm{INR}(t) = \mathrm{INR}_0\,\mathrm{PT}(t)^{\,\mathrm{ISI}}, \tag{20}$$

where INR_0 is the baseline INR and ISI is the sensitivity index of the assay.

Patient heterogeneity. Patients differ in CYP2C9 metabolism class, VKORC1 sensitivity, age, clearance, absorption, effect-site kinetics, drug sensitivity, baseline vitamin K, adherence, and the dose they need for maintenance. A clinic-level bias further shifts dosing decisions.

Behaviour policy. The dose is drawn from a softmax over the four dose levels. Its logits depend mainly on the current INR, its distance from the therapeutic range, and its trend. They also depend on the effect-site concentration, recent dose load, adherence, age, maintenance-dose class, and clinic bias. The confounding level γ scales the INR-dependent terms, so a larger γ ties dosing more closely to the patient's current state and strengthens time-varying confounding.

State and outcome. The outcome is $\mathrm{y}_t = \mathrm{INR}_t$. The model also sees $d_x = 9$ covariates: concentration, effect-site concentration, factors II and VII, vitamin K, the dose load over the past seven days, CYP class, VKORC1 class, and normalized age.

Queries. A query asks for a patient's outcome under one treatment plan at one point in their timeline. The prediction point slides forward one step at a time, starting once $T_{\min} = 10$ steps have been observed. At each point the simulator is rerun under all $n_a = 4$ doses for a one-step query and under $2H = 10$ plans drawn at random from $\mathcal{A}^H$ for a horizon query. A horizon query must fit entirely inside the $T_{\max} = 60$ step timeline, so its prediction point stops H steps before the end. This leaves 49 one-step and 45 horizon points, and a task contributes $n_a \times 49 = 196$ one-step and $2H \times 45 = 450$ horizon-5 queries.

C.3 HIV

This domain uses the HIV treatment ODE of Adams et al. (2004), as implemented in the WhyNot suite (Miller et al., 2020). It describes how the immune system and the virus change over time during antiretroviral treatment, and interventional targets come from altering the treatments a patient receives in the future.

Treatment. Each regimen is defined by two efficacies, ϵ_1 for RTI and ϵ_2 for PI. Treatment $\mathrm{a}_t = 0$ gives neither drug, $(\epsilon_1, \epsilon_2) = (0, 0)$; $\mathrm{a}_t = 1$ gives PI alone, $(0, 0.3)$; $\mathrm{a}_t = 2$ gives RTI alone, $(0.7, 0)$; and $\mathrm{a}_t = 3$ combines both, $(0.7, 0.3)$.

Dynamics. Six quantities make up the state: the free virus V, the immune response E, and two kinds of target cells, each split into a healthy population (P_1, P_2) and an infected one (P_1^*, P_2^*).

Their evolution is

$$\dot{P}_1 = \lambda_1 - d_1 P_1 - (1-\epsilon_1)\, k_1 V P_1, \tag{21}$$

$$\dot{P}_1^* = (1-\epsilon_1)\, k_1 V P_1 - \delta P_1^* - \eta_1 E P_1^*, \tag{22}$$

$$\dot{P}_2 = \lambda_2 - d_2 P_2 - (1-f\epsilon_1)\, k_2 V P_2, \tag{23}$$

$$\dot{P}_2^* = (1-f\epsilon_1)\, k_2 V P_2 - \delta P_2^* - \eta_2 E P_2^*, \tag{24}$$

$$\dot{V} = (1-\epsilon_2)\, N\delta\,(P_1^* + P_2^*) - cV - \big[(1-\epsilon_1)\,\rho_1 k_1 P_1 + (1-f\epsilon_1)\,\rho_2 k_2 P_2\big] V, \tag{25}$$

$$\dot{E} = \lambda_E + \frac{b_E\,(P_1^* + P_2^*)}{P_1^* + P_2^* + \kappa_b}\, E - \frac{d_E\,(P_1^* + P_2^*)}{P_1^* + P_2^* + \kappa_d}\, E - \delta_E E. \tag{26}$$

Patient heterogeneity. Variation across patients comes from four sources: random perturbations of the ODE parameters, a per-patient scaling of each drug's efficacy, individual thresholds on viral load and immune response, and a patient-specific degree of treatment aggressiveness.

Behaviour policy. At each step, a severity score is formed from the current virus and immune levels, each transformed as $\log_{10}(1+\cdot)$. This score is multiplied by γ and combined with a term that favours staying on the previous treatment, and a softmax over the four categories gives the probability of the next one. The higher γ is, the more the choice of regimen reflects the patient's biological state.

State and outcome. The model observes each of the six quantities after a $\log_{10}(1+\cdot)$ transform. Free virus on this scale, $\mathrm{y}_t = \log_{10}(1+V_t)$, is the outcome, and the other five quantities form the covariates ($d_x = 5$).

Queries. A query asks for a patient's outcome under one treatment plan at one point in their timeline. The prediction point slides forward one step at a time, starting once $T_{\min} = 10$ steps have been observed and running through the $T_{\max} = 60$ step timeline. This gives 49 one-step and 50 horizon points. At each point, the simulator is rerun under all $n_a = 4$ regimens for a one-step query and under $2H = 10$ plans drawn at random from $\mathcal{A}^H$ for a horizon query. A task therefore contributes $n_a \times 49 = 196$ one-step and $2H \times 50 = 500$ horizon-5 queries.

C.4 MIMIC-III

This benchmark is built from hourly ICU stays in MIMIC-III (Johnson et al., 2016; Pollard et al., 2026) using a MIMIC-Extract-style pipeline (Wang et al., 2020), with each stay treated as one sequence. The 10-dimensional state consists of diastolic and mean blood pressure, oxygen saturation, heart rate, respiratory rate, Glasgow Coma Scale total, glucose, creatinine, bicarbonate and sodium. Static covariates $\mathbf{s}$ are derived from demographics. Treatment combines vasopressor use and mechanical ventilation as $\mathrm{a}_t = \mathbb{I}\{\mathrm{vaso}_t\} + 2\,\mathbb{I}\{\mathrm{vent}_t\}$. The outcome is diastolic blood pressure. Because interventional outcomes are unobservable, all MIMIC-III queries are factual.

Queries. Queries are generated by sliding the prediction point forward as in the simulated domains, from $T_{\min} = 10$ to near the end of the record. The difference is that there is no simulator: the only outcome ever observed is the one that followed the treatment actually given, so nothing can be re-run under an alternative treatment or plan and each point yields one factual query rather than n_a or $2H$ of them. A task therefore contributes at most 49 one-step and 50 horizon-5 queries. Prediction points whose outcome was not recorded are dropped.

D BASELINES

This appendix describes the comparison methods. The six dataset-specific estimators and the CLPFN evaluation all follow Zare et al. (2026). Every outcome-prediction number in Table 1 other than ours is taken directly from that work. The covariate-forecasting results for G-Net and CLPFN are new; we produced them with the same code and protocol (Section D.1). The six estimators span the principal ways of handling time-varying confounding: inverse-probability weighting (MSM,

Table 8: Overview of the comparison methods.

Method	Backbone	Confounding adjustment	Training objective	Covariate forecasts
MSM	Linear / ridge regression	Inverse-propensity weights (logistic)	Weighted per-horizon regression	–
RMSN	LSTM encoder–decoder	Inverse-propensity weights (RNN)	Weighted MSE	–
G-Net	LSTM	g-computation	Outcome + covariate MSE	native
CRN	LSTM encoder–decoder	Adversarial balancing	Outcome MSE + treatment CE	–
CT	Multi-stream transformer	Adversarial balancing	Outcome MSE + treatment CE	–
GT	Multi-stream transformer	g-computation	Outcome MSE	–
CLPFN	pre-trained transformer (PFN)	Learned from the prior	Synthetic NLL	one target per run

RMSN) (Robins et al., 2000; Lim, 2018), neural g-computation (G-Net, GT) (Li et al., 2021; Hess et al., 2024), and adversarially balanced representations (CRN, CT) (Bica et al., 2020; Melnychuk et al., 2022). Table 8 gives an overview.

All methods use the timing convention of Section 3.2. The history is available up to and including step T. The first planned treatment $\bar{a}_T$ produces the state at $T{+}1$. The forecast for step $T{+}h$ may use only $\bar{a}_T, \ldots, \bar{a}_{T+h-1}$.

Scaling and scoring. Continuous inputs of the baselines are standardized with statistics of the context trajectories in the corresponding benchmark file. Scores are reported as RMSE on a common scale that is likewise computed from the context set only. The PFNs normalize internally with their own context statistics. Their predictions are therefore first mapped back to raw units and then onto the common scale, so that all methods are scored identically.

Model selection. The six estimators are tuned with context data only, following the procedure of Zare et al. (2026) without modification. The search grid is given in Table 9. We adopt it unchanged so that our numbers remain directly comparable to the baseline results reported there. In all cases the selected configuration is retrained on the full context set before any query is scored, and query labels never influence the selection. The two PFNs receive no such treatment: both are used as frozen pre-trained networks, with no parameter updates, no validation split and no hyperparameter choices on the target domain. For the CLPFN, we use the released checkpoint (`https://huggingface.co/Amirhossein-Zare/causal-long-pfn`) without modification.

D.1 Covariate forecasting

Only two comparison methods can be scored on covariates. G-Net produces covariate forecasts as part of its standard rollout, and we evaluate those forecasts directly. CLPFN has a single target column, so we query it once per covariate, labelling covariate j as the target variable and the original outcome as one of the remaining variates. This relabelling is not a departure from the evaluation protocol: in these domains every time-varying variable is endogenous, driven by the past of the other variables and by the assigned treatments. CLPFN therefore still conditions on the full system, and a domain requires d_x separate forward passes per query. MTS-CFM, by contrast, predicts all d_x covariates and the outcome in one pass. For all three models, covariate errors are computed per variable on the context-normalized scale.

E Additional Results

This appendix breaks the aggregate covariate results of Table 2 down by variable, and reports the distributional metrics referenced in Section 5.2. Tables 10–12 give per-covariate RMSE on HIV, MIMIC-III and Warfarin for MTS-CFM, CLPFN and G-Net; the best value in each row is in bold. MTS-CFM is best on 16 of 23 covariates at one step and on 20 of 23 at horizon-5.

Table 9: Search grids for the dataset-specific baselines, following Zare et al. (2026). Candidates are sampled at random from the listed values and ranked on a held-out split of the context set. The PFNs are absent because they are not tuned on the target domain.

Method	Model capacity	Training and regularization
MSM	Outcome regressor: linear or ridge ($\alpha \in \{0.1, 1, 10\}$); history of 2 lags.	Logistic propensities ($C = 10^6$, at most 500 iterations); stabilized weights clipped to the 1% and 99% quantiles of the context set.
RMSN	1 or 2 recurrent layers. Encoder and decoder widths come from a grid that scales with the input size: $C_{\text{hist}} = d_x + 8$ and $C_{\text{dec}} = 8$ are multiplied by $\{0.5, 1, 2, 4\}$, rounded to multiples of 16, clipped to $[32, 160]$, and merged with $\{32, 48, 64, 96, 128\}$.	Dropout $\{0.1, \ldots, 0.5\}$; learning rates of the propensity, encoder and decoder networks each in $\{10^{-2}, 10^{-3}, 10^{-4}\}$; encoder batch $\{64, 128, 256\}$; decoder batch $\{256, 512, 1024\}$; gradient clip $\{0.5, 1, 2, 4\}$.
G-Net	Hidden width $\{48, 64, 96, 128\}$; representation width $\{48, 64, 96\}$; 1 or 2 recurrent layers.	Dropout $\{0.05, 0.1, 0.2\}$; learning rate $\{10^{-3}, 3{\times}10^{-4}\}$; batch $\{32, 64\}$; epochs $\{80, 120\}$; covariate loss weight $\lambda_x \in \{0.15, 0.25, 0.35\}$.
CRN	1 or 2 recurrent layers. Hidden, balanced and fully connected widths come from an input-scaled grid built from $C_{\text{hist}} = d_x + 10$ and $C_{\text{dec}} = 10$, as for RMSN. It is merged with $\{32, 48, 64, 96, 128\}$ for the hidden and balanced widths and with $\{32, 64, 96, 128, 192\}$ for the fully connected width.	Dropout $\{0.1, \ldots, 0.5\}$; encoder and decoder learning rates each in $\{10^{-2}, 10^{-3}, 10^{-4}\}$; encoder batch $\{64, 128, 256\}$; decoder batch $\{256, 512, 1024\}$; gradient clip $\{0.5, 1, 2\}$.
CT	Layers $\{2, 3, 4\}$; heads $\{2, 4\}$; sequence, balanced and fully connected widths each in $\{64, 96, 128\}$.	Dropout $\{0.05, 0.1, 0.2\}$; learning rate $\{10^{-3}, 3{\times}10^{-4}, 10^{-4}\}$; weight decay $\{10^{-5}, 10^{-4}, 10^{-3}\}$; batch $\{32, 64\}$; gradient clip $\{0.5, 1\}$; treatment loss weight $\{0.05, 0.1, 0.2\}$.
GT	Layers $\{2, 3, 4\}$; heads $\{2, 4\}$; model and balanced widths each in $\{32, 48, 64, 96, 128\}$; fully connected width $\{32, 64, 96, 128, 192\}$.	Dropout $\{0.1, 0.2, 0.3\}$; learning rate $\{10^{-3}, 10^{-4}, 10^{-5}\}$; weight decay $\{10^{-5}, 10^{-4}, 10^{-3}\}$; batch $\{16, 32, 64\}$.

Table 10: Per-covariate prediction results on HIV. Normalized RMSE, lower is better.

	MTS-CFM		CLPFN		G-Net	
Covariate	One-step	Horizon-5	One-step	Horizon-5	One-step	Horizon-5
P_1	**0.038±0.070**	**0.195±0.298**	0.051±0.089	0.309±0.504	0.082±0.105	0.296±0.405
P_1^*	**0.053±0.031**	**0.203±0.112**	0.067±0.041	0.262±0.162	0.103±0.055	0.221±0.131
P_2	**0.038±0.027**	**0.192±0.152**	0.044±0.030	0.268±0.176	0.092±0.061	0.228±0.125
P_2^*	**0.028±0.010**	**0.133±0.073**	0.063±0.030	0.302±0.188	0.096±0.058	0.244±0.153
E	**0.004±0.003**	**0.024±0.013**	0.012±0.007	0.086±0.051	0.063±0.059	0.170±0.118

Table 11: Per-covariate prediction results on MIMIC-III. Normalized RMSE, lower is better.

	MTS-CFM		CLPFN		G-Net	
Covariate	One-step	Horizon-5	One-step	Horizon-5	One-step	Horizon-5
Mean blood pressure	**0.616±0.190**	**0.666±0.237**	0.662±0.368	0.720±0.265	0.633±0.236	0.737±0.289
Oxygen saturation	**0.534±0.226**	**0.534±0.279**	0.562±0.357	0.621±0.450	0.554±0.269	0.601±0.339
Heart rate	0.415±0.166	0.517±0.224	**0.364±0.147**	**0.516±0.233**	0.403±0.159	0.580±0.216
Respiratory rate	**0.660±0.225**	**0.702±0.215**	0.666±0.242	0.729±0.226	0.676±0.236	0.759±0.257
Glasgow coma scale	0.155±0.224	**0.224±0.281**	**0.119±0.178**	0.228±0.279	0.199±0.190	0.317±0.260
Glucose	0.394±0.242	**0.537±0.313**	**0.313±0.219**	0.598±0.476	0.349±0.212	0.576±0.343
Creatinine	**0.051±0.067**	**0.083±0.086**	0.051±0.087	0.106±0.160	0.171±0.248	0.204±0.179
Bicarbonate	0.179±0.094	0.316±0.148	**0.136±0.073**	**0.304±0.149**	0.229±0.189	0.382±0.206
Sodium	0.180±0.105	0.324±0.166	**0.139±0.086**	**0.315±0.174**	0.238±0.231	0.375±0.171

Table 12: Per-covariate prediction results on Warfarin. Normalized RMSE, lower is better.

	MTS-CFM		CLPFN		G-Net	
Covariate	One-step	Horizon-5	One-step	Horizon-5	One-step	Horizon-5
Concentration	**0.005±0.021**	**0.019±0.070**	0.018±0.070	0.075±0.222	0.046±0.128	0.090±0.236
Effect site	**0.002±0.004**	**0.008±0.026**	0.008±0.020	0.047±0.130	0.046±0.129	0.087±0.235
Factor II	**0.013±0.004**	**0.074±0.018**	0.027±0.007	0.166±0.051	0.140±0.129	0.237±0.236
Factor VII	**0.051±0.013**	**0.247±0.064**	0.052±0.011	0.334±0.085	0.123±0.109	0.341±0.184
Vitamin K	0.102±0.024	**0.354±0.084**	**0.069±0.014**	0.396±0.098	0.196±0.125	0.443±0.170
7-day dose load	0.051±0.009	**0.129±0.023**	**0.038±0.005**	0.144±0.021	0.162±0.182	0.270±0.188
CYP2C9	**0.002±0.001**	**0.007±0.009**	0.008±0.002	0.021±0.007	0.158±0.230	0.210±0.264
VKORC1	**0.003±0.002**	**0.005±0.004**	0.008±0.005	0.022±0.015	0.174±0.305	0.221±0.255
Age	**0.003±0.001**	**0.008±0.009**	0.007±0.004	0.030±0.018	0.134±0.220	0.182±0.197