TRAINING AND EVALUATING CAUSAL FORECASTING MODELS FOR TIME-SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep learning time-series models are often used to make forecasts that inform downstream decisions. Since these decisions can differ from those in the training set, there is an implicit requirement that time-series models will generalize outside of their training distribution. Despite this core requirement, time-series models are typically trained and evaluated on in-distribution predictive tasks. We extend the orthogonal statistical learning framework to train causal time-series models that generalize better when forecasting the effect of actions outside of their training distribution. To evaluate these models, we leverage Regression Discontinuity Designs popular in economics to construct a test set of causal treatment effects.

1 INTRODUCTION

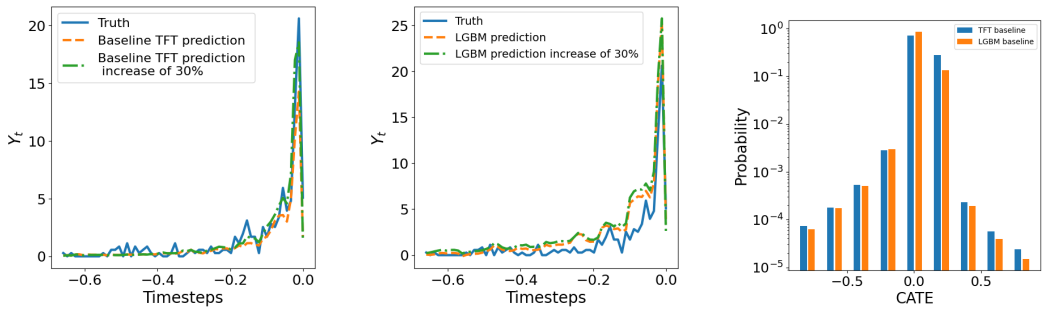
Deep learning models have recently seen major improvements, including in time-series modelling Liu et al. (2024); Nie et al. (2023); Chen et al. (2023); Das et al. (2024); Zhang & Yan (2023). Time-series models are often used to make forecasts that inform decisions, such forecasting demand for goods or transportation to optimize pricing, or forecasting health markers to optimize treatment decisions Makridakis et al. (2022); Nowroozilarki et al. (2021); Makridakis et al. (2023).

The decision making procedures that leverage time-series forecasts typically optimize for a target outcome, which leads to actions that differ from those observed in the training data. This results in a key implicit requirement on time-series models: that their predictions will generalize to action values outside of the observational distribution. Despite this requirement, the only complex tasks on which time-series forecasting models are trained and evaluated are in-distribution predictive tasks Liu et al. (2024); Melnychuk et al. (2022); Lim (2018); Nie et al. (2023); Chen et al. (2023); Das et al. (2024); Zhang & Yan (2023); Nowroozilarki et al. (2021); Makridakis et al. (2022; 2023).

Asking models to predict on out-of-distribution actions can drastically degrade performance. Figure 1 shows predictions on a passenger rail demand forecasting task, in which the model predicts the daily number of seats sold at a given price for a tain. The price is the action that the forecast will help optimize. We fit a well tuned LightGBM Ke et al. (2017) model, and an improved version of the Temporal Fusion Transformer (TFT) Lim et al. (2020). Both models have comparable performance for in-distribution prices (Fig. 1a, 1b, details in §4). We then predict demand under a 30% price increase (green line). Of course, we would expect higher prices to cause a *decrease* in demand. Instead, both models predict an *increase* in demand. Figure 1c confirms that this effect is widespread, affecting 13% and 28% of the predictions for the LGBM and TFT models, respectively.

This incorrect behaviour appears because models heavily rely on the information contained in in-distribution actions for their predictions. Indeed, in the collected data actions are set by an informed decision maker (an operator of the train company who knows when demand will be high from experience), which confounds observed actions and associated outcomes. In our data, we do observe higher prices when demand is high (e.g., during holidays), and low prices when demand is low. This signal breaks when predicting on out-of-distribution actions (they are not chosen by an informed decision maker), leading to inaccurate predictions.

In this work, we develop a training and evaluation procedure for causal forecasting time-series models. Causal models aim to learn the causal relationship between the action (e.g., price) and its effect on a specific outcome (e.g., demand), conditioned on other observational features. That is, they aim to capture the change in outcome caused by a change in action, and not just predict the outcome from



(a) TFT (b) LGBM (c) Treatment effects distribution

Figure 1: (a), (b): Daily demand forecast for one train, and its change under a price increase. Blue: ground truth; orange: the model's forecast for observed prices; green: the forecast under a 30% price increase. Both models predict an increase in demand under increased prices. (c): Distribution of predicted changes of demand in the test set, for a change from the observed price to the next possible price, normalized by the price change.

an in-distribution observed treatment. As a result, causal models generalize better when forecasting the effect of actions outside of their training distribution.

A few causal forecasting models have been proposed, but they use approaches that do not map to well defined causal effects Bica et al. (2020a;b); Melnychuk et al. (2022), do not correct for regularization bias Li et al. (2020a); Bica et al. (2020a;b); Melnychuk et al. (2022); Lim (2018), or take inspiration from causal frameworks for heuristics Brodersen et al. (2015); Schultz et al. (2024). In this work, we leverage the orthogonal statistical learning framework of Foster & Syrgkanis (2023) to learn a causal forecasting model. We extend this framework to a well-defined time-series causal problem and to high-dimensional treatments, and instantiate it on top of state-of-the-art time-series architectures to support complex use-cases.

Evaluating causal forecasting models is even more challenging, as outcomes under alternative treatments are by definition unobserved. Existing work thereby rely on simple simulations with known counterfactuals Bica et al. (2020a;b); Li et al. (2020a); Lim (2018), or in-distribution forecasting performance Lim (2018); Melnychuk et al. (2022). The former is often too simplistic to really compare models, while the later can be misleading and inflate the performance of non-causal models. We show details of this effect in §2.1.

To evaluate forecasting models on causal tasks, we take inspiration from Regression Discontinuity Designs (RDDs) popular in econometrics Hahn et al. (2001); Imbens & Lemieux (2008), to construct a test set of causal effects estimated under different assumptions than the typical model-fitting assumption. We can then compare the treatment effects estimated by forecasting models using traditional evaluation metrics. We train and evaluate our causal forecasting models on two relevant tasks: demand forecasting using a proprietary dataset of passenger rail prices and demand over time; and the public health dataset MIMIC-III Johnson et al. (2016). We show that, compared to traditional time-series forecasting models and a state-of-the-art causal model, our causal forecasting models predict causal effects that are 36% and 1% closer to those estimated with RDDs, respectively.

2 Causal Forecasting with Orthogonal Learning

In this section and next (§3), we present our approach using our demand forecasting task for passenger rails as a running example (see §4.1 for more details on the data and task). We emphasize that the theory is general and applies to other time-series settings as well. We demonstrate this applicability in §4, where we evaluate our models on our demand forecasting task using a proprietary dataset, as well as on the public MIMIC-III health forecasting dataset Johnson et al. (2016).

We abstract forecasting tasks as a dataset of N observed time-series (N trains) indexed by n, with time indexed by t. Each time-series n consists in stationary context features S^n, temporal (non-stationary) features measured at each time step $X^n_{1:t}$, an action at each time step $A^n_{1:t}$ (the price), and real valued outcomes $Y^n_{1:t}$ ($Y^n_t \in \mathbb{R}$, the demand). The objective is to predict each outcome Y^n_t

using static, temporal, and treatment features, as well as past outcomes up to a time $\tau < t$ ($Y^n_{\tau+1:t-1}$ is unknown). We thus seek to learn $\hat{Y}^n_t \triangleq f(S^n, X^n_{1:t}, A^n_{1:t}, Y^n_{1:\tau})$, such that $\hat{Y}^n_t$ is close to Y^n_t.

We next formalize causal effects and introduce orthogonal learning theory that we leverage in our models (§2.1). We then extend orthogonal learning to time-series models (§2.2), and instantiate this theory with deep learning architectures (§2.3).

2.1 Background: Causal Treatment Effects and Orthogonal Learning

To formalize the importance of causality in forecasting for optimizing decisions, we first need to introduce treatment effects. In the reminder of this paper, we use the potential outcomes' model of causal inference Holland (1986); Imbens & Rubin (2015), and flexible data models Robinson (1988); Nie & Wager (2020); Chernozhukov et al. (2018).

Potential Outcomes represent the values of an outcome under different possible actions (also commonly called treatments or decisions in the literature). Consider a two actions setting: $A = a$ or $A = b$. For a time-series n and time step t, the potential outcomes $Y^n_t(a)$ and $Y^n_t(b)$ are the value of the demand Y^n_t we would observe under each action (price). Of course, we only ever observe one potential outcome, as $Y^n_t = \mathbb{1}_{A^n_t=a} Y^n_t(a) + \mathbb{1}_{A^n_t=b} Y^n_t(b)$ where $\mathbb{1}$ is the indicator function.

Treatment Effects represent the causal effect of a change in action (price) on the outcome of interest (demand). Formally, the treatment effect of going from action a to b, for time-series n, at time step t, is $Y^n_t(b) - Y^n_t(a)$. The fundamental challenge of causal inference is that we can only ever observe one potential outcome, for the treatment we chose. As a result, we cannot train a supervised treatment effects model. However, we can estimate expected causal effects under unconfoundedness, the main assumption in the literature which we make when learning our models:

Assumption 1 (Unconfoundedness)**.** *Consider a set of possible treatments $\mathcal{T}$. We say that the time-series $(Y_t(A_t), A_t, S, X_t)$ are (conditionally) uncounfounded if, for every time-step $t > \tau$:*

$$\{Y_t(A_t),\ A_t \in \mathcal{T}\} \perp\!\!\!\perp A_t \mid S,\ X_{1:t},\ A_{1:\tau},\ Y_{1:\tau}$$

Intuitively, this assumption means that in sub-populations defined by the conditioning variables, the groups receiving each treatment are comparable (have the same potential outcomes distribution). Unconfoundedness is verified by construction in conditionally randomized trials, as treatments are assigned independently of potential outcomes. It is also verified when context variables $S,\ X_{1:t},\ A_{1:\tau},\ Y_{1:\tau}$ include all pre-treatment variables that are common causes of A^n and Y^n Pearl (2009). In our running example, this consists in all variables that influence both demand and pricing decisions, such as the cities of arrival and departure, day of the trip, and more. Under Assumption 1, a well known identifyability result holds (see e.g., Imbens & Rubin (2015); Pearl (2009)):

Proposition 1 (CATE Identifyablity)**.** *Under Assumption 1, we can identify the Conditional Average Treatment Effect (CATE) as follows:*

$$\begin{aligned} \textit{CATE} &\triangleq \mathbb{E}\big[Y_t(b) - Y_t(a) \mid S,\ X_{1:t},\ A_{1:\tau},\ Y_{1:\tau}\big] \\ &= \mathbb{E}\big[Y_t \mid A_t = b,\ S,\ X_{1:t},\ A_{1:\tau},\ Y_{1:\tau}\big] - \mathbb{E}\big[Y_t \mid T_t = a,\ S,\ X_{1:t},\ A_{1:\tau},\ Y_{1:\tau}\big]. \end{aligned}$$

This important result shows that we can estimate the Conditional Average Treatment Effect (CATE) from conditional expectations $\mathbb{E}\big[Y_t \mid A_t = a,\ S,\ X_{1:t},\ A_{1:\tau},\ Y_{1:\tau}\big]$. This is a familiar quantity to ML practitioners, as a flexible forecasting model $f : A, S, X, A_{1:\tau}, Y_{1:\tau} \to Y$ trained with a mean-squared error will converge to this conditional expectation in the limit of infinite data. Estimating the CATE $\mathbb{E}\big[Y_t(b) - Y_t(a) \mid S,\ X_{1:t},\ A_{1:\tau},\ Y_{1:\tau}\big]$ is then as simple as predicting on the two treatment values $f(T_t = b, \ldots) - f(T_t = a, \ldots)$. Unfortunately, practice is more challenging.

The need for causal forecasting arises when fitting complex forecasting models with finite sample sizes. In observational data collected from regular interactions, such as observed demand for passenger rail seats during regular operation, some treatments will be over-represented. For instance, in our data high prices are over-represented when demand is high, and low prices when demand is low. Of course, we expect higher prices to cause a decrease in demand, but the correlation in the data is flipped due to the transportation company operators using their experience and domain knowledge about demand to set higher prices when underlying demand is high, thereby increasing revenue.

When trained on finite data with such biased treatment assignments, forecasting models will extrapolate based on feature and treatment values. This introduces bias in treatment effect estimates when models are not flexible enough to represent the true function, or due to regularization (both explicit regularization such as weight decay, or deep learning models' implicit regularization (Soudry et al., 2018; Ji et al., 2020)). This effect, called regularization bias, happens even under simple treatment effect functional forms (Chernozhukov et al., 2018). In practice, this is exactly what we observe when fitting ML models on our demand forecasting task, as shown in §1 and Figure 1.

Algorithm 1 Sample Splitting Learning Algorithm from Foster & Syrgkanis (2023)

Input: Sample set $S = z_1, \dots, z_n$
Split S into subsets $S_1 = z_1, \dots, z_{\lfloor n/2 \rfloor}$, and $S_2 = S \backslash S_1$
Fit g for g_0 on S_1
Fit θ for θ_0 on S_2 using g for g_0 in the loss
Return g, θ

Orthogonal Learning (Foster & Syrgkanis, 2023) is a recent learning theory framework to study the convergence rate of models with nuisance parameters, which can be used to estimate CATE models. The framework is very general: we next describe the core results relevant our work, that we later extend to our setting (§2.2). The results for treatment effect estimation consider a standard binary action model (Robinson, 1988; Nie & Wager, 2020; Chernozhukov et al., 2018), with observations $z = (W, Y, A)$ from distribution $\mathcal{D}$, and $A \in \{0, 1\}$, such that:

$$Y = A \cdot \theta_0(W) + f_0(W) + \epsilon_1, \qquad \mathbb{E}[\epsilon_1 | W, A] = 0 \tag{1}$$
$$A = e_0(W) + \epsilon_2, \qquad \mathbb{E}[\epsilon_2 | W] = 0 \tag{2}$$

Further define $m_0(w) \triangleq \mathbb{E}[Y|W = w]$. We call $g_0 \triangleq (e_0, m_0)$ the nuisance parameters, required to learn the CATE θ_0 but not direct objects of interest in this framework. In this model $\theta_0(W)$ is the CATE (as it is turned on and off by A, see Proposition 1). This data model is quite flexible, as the CATE depends arbitrarily on the features W through θ_0, as do the outcome $Y \in \mathbb{R}$ and action A through f_0, e_0. Note that under Assumption 1, this decomposition is without loss of generality.

Calling θ, e, m models for θ_0, e_0, m_0, respectively, we can define the residualized loss (R-loss) $l(\cdot)$, and its population equivalent $L_{\mathcal{D}}(\cdot)$, as:

$$l(\theta, m, e; z) = \Big(Y - m(W) - \big(A - e(W)\big)\theta(W)\Big)^2 \;\; ; \;\; L_{\mathcal{D}}(\theta, m, e) = \mathbb{E}_{z\sim\mathcal{D}}[l(\theta, m, e; z)].$$

Now consider a model g for g_0 learned on sample S, such that $\|g - g_0\|_{L_2} \triangleq \sqrt{\mathbb{E}_{z\sim\mathcal{D}}\|g(w) - g_0(w)\|_2^2} \leq \text{Rate}_{\mathcal{D}}(S, \delta)$ w.p. at least $1 - \delta$. Since $m_0(w) = \mathbb{E}[Y|W = w]$ and $e_0(w) = \mathbb{E}[A|W = w]$, these can be models that minimize the mean squared error.

Further consider an estimation algorithm of model θ for θ_0 which, given a model g for nuisance parameters, outputs $\theta \in \Theta$ using sample S (e.g., by minimizing the empirical loss $l(\cdot)$ over S) such that $L_{\mathcal{D}}(\theta, g) - L_{\mathcal{D}}(\theta^*, g) \leq \text{Rate}_{\mathcal{D}}(S, \delta; \theta, g)$, where $\theta^* \triangleq \arg\min_{\theta\in\Theta} L_{\mathcal{D}}(\theta, m_0, e_0)$ is the best predictor for θ_0 in class Θ. A low risk $L_{\mathcal{D}}(\theta, g) - L_{\mathcal{D}}(\theta^*, g)$ does not indicate a good model θ, as the loss l is computed with g, and not g_0 the true nuisance parameter. However, Foster & Syrgkanis (2023) show (in Theorem 2, and Appendix J.2, with a constant we derive in Appendix B) that:

Proposition 2 (Orthogonal Learning for Binary Treatment Effect)**.** *Consider a class of functions Θ where $\forall W,\ \forall \theta \in \Theta :\ |\theta(W)| \leq M$. Fitting g and θ with the procedures above using Algorithm 1, we have that with probability at least $1 - \delta$:*

$$\mathbb{E}\big[\big(A - e_0(W)\big)\big(\theta(W) - \theta^*(W)\big)\big]^2 \leq L_{\mathcal{D}}(\theta, g_0) - L_{\mathcal{D}}(\theta^*, g_0)$$
$$\leq \textit{Rate}_{\mathcal{D}}(S_2, \delta/2; \theta, g) + 2(1 + M^2) \cdot \textit{Rate}_{\mathcal{D}}(S_1, \delta/2)^2.$$

There are three key takeaways. **(1)** we can get a CATE model θ close to the best possible predictor in the class (which can be θ_0 if $\theta_0 \in \Theta$) for all values W with treatment variation around the mean. This is similar to the positivity assumption ($\forall A, W : 1 > \mathbb{P}(A|W) > 0$) pervasive in causal

inference results. **(2)** the rate at which θ converges to θ^* under estimated nuisance parameters g is sufficient to bound the error under the true parameters g_0. **(3)** the rate of estimation of the nuisance parameters is squared: Algorithm 1 is tolerant of errors in estimating complex nuisance parameters. For instance, a rate of convergence of $1/\sqrt{N}$ for θ, is preserved if g converges as $o(1/N^{\frac{1}{4}})$.

2.2 Orthogonal Learning for Time Series Models

We adapt the orthogonal statistical learning framework described above by making two key extensions: defining daily treatment effects with an observation cut-off; and extending the R-loss to categorical and linear effects with various encodings for θ's predictions.

Treatment effects in time series. The first step is to formalize the treatment effects to predict, and the context to predict them with. In this paper, we focus on *daily treatment effects* defined in Proposition 1. This modelling choice lets us focus on single-day treatments, though supporting sequences of treatments is an interesting avenue for future work. Focusing on a time-step t, this implies that $Y = Y_t$ in Equation 1 and $A = A_t$ in Equation 2.

Defining the relevant feature set (W in Eq. 1, 2) is more tricky. Recall from §2 that our time series consist in static (S) and temporal ($X_{1:t}$) features. These features are naturally included ($W \supset (S, X_{1:t})$). In time-series, results from past time steps, such as past prices and associated demands $(A_{1:t-1}, Y_{1:t-1})$, are typically very predictive. However, both the uncondoundedness Assumption 1 and the orthogonal leaning data model (Eq. 1, 2) implicitly assume that features W follow the observational distribution at training and prediction time. This assumption is not verified when models are used to influence past decisions, and $A_{<t}, Y_{<t}$ cannot be included in X. This is an important point: observational past treatment are very predictive of future demand, but confound the daily treatment effect and are the cause of reversed elasticity predictions we observed (§2.1, Figure 1). To avoid this confounding, we introduce a cut-off τ after which treatments can deviate from the distribution. That is, in time-steps $1:\tau$ the model is not used to affect treatments, while for $t > \tau$, the treatment A_t can differ from the training distribution. We then set the features as $W = (S, X_{1:t}, A_{1:\tau}, Y_{1:\tau}) \triangleq W_t$, which we use as input for our daily treatment effect of A_t on Y_t.

High dimensional effects. The other extension we require is support for higher-dimensional treatments. We consider both categorical and linear treatments and treatment effects. We start with categorical treatments, for which we propose two different encodings. In our demand forecasting dataset, possible prices are discrete (in our health application too), following a large but tractable set of possible values (about thirty possible prices) of size $d \triangleq |\mathcal{A}|$. We extend the approach of Robinson (1988); Nie & Wager (2020); Foster & Syrgkanis (2023) (Equations 1, 2) to model each time-step t as follows:

$$Y_t = A_t^T \theta_0(W_t) + f_0(W_t) + \epsilon_1, \qquad \mathbb{E}[\epsilon_1 | W_t, A_t] = 0 \tag{3}$$

$$A_t = e_0(W_t) + \epsilon_2, \qquad \mathbb{E}[\epsilon_2 | W_t] = \mathbf{0}_d \tag{4}$$

where A_t is a d-dimensional column vector encoding the treatment, $\theta_0(W_t) \in \mathbb{R}^d$ outputs a column vector treatment effect, and $\mathbf{0}_d$ is the d-dimensional zero vector. The R-loss becomes:

$$l(\theta, m, e, z_t) = \Big(Y_t - m(W_t) - \big(A_t - e(W_t)\big)^T \theta(W_t)\Big)^2. \tag{5}$$

We consider two encodings for the categorical treatment A_t and treatment effect $\theta(W_t)$. In the *one-hot encoding*, A_t is a one-hot vector of the treatment. As a result, the i^{th} dimension of the treatment effect model encodes the CATE compared to f_0. That is, $\theta(W_t)_i = \mathbb{E}\big[Y_t(a_i) - f_0(W_t) \mid W_t\big]$.

In the *cumulative encoding*, the treatment model encodes the treatment effect of incremental price changes, such that the cumulative predictions encode the CATE: $\sum_{j=1}^{i} \theta(W_t)_j = \mathbb{E}\big[Y_t(a_i) - f_0(W_t) \mid W_t\big]$. In this case, A_t is a vector of ones in dimensions $1:i$, and zeros after. While we do not do it in this work, the cumulative encoding can be useful to encode constraints, such as fixing the sign of incremental effects of price changes.

Given true categorical treatments and our flexible deep-learning models, we can reasonably assume that $\theta_0 \in \Theta$, and thus $\theta^* = \theta_0$. With enough treatment variation, our model will converge to the true CATE. In practice however, demand forecasting datasets only see local variation around a typical

price for any particular value of features W_t. As a result, a small risk under the R-loss only ensures an accurate θ around likely prices. The model is under-constrained further from typical prices, where it will not learn relevant treatment effects. This is a fundamental limitation of any causal approach, as local variations around a typical price implies a lack of positivity for most dimensions of $\theta(W_t)$.

In such cases, adding structure to the treatment effect model is practically useful. To this end, we also consider a *linear encoding* for the treatment effect. Concretely, we use the data model of Equations 3 and 4, where A is a one-dimensional, real treatment value (the price at which we want to predict demand), and the treatment effect $\theta(W_t)$ is a multiplicative coefficient. This creates a linear effect model, where the linear coefficient depends on W_t in a flexible way. As a result, the real treatment effect is probably not in the model class ($\theta_0 \notin \Theta$), and we converge to θ^*. In exchange, estimations under this model can be much more stable under moderate treatment variability.

All encodings are compatible with the following novel result:

Proposition 3 (Orthogonal Learning for Categorical Treatment Effects)**.** *Consider a class of functions $\Theta : \mathcal{W} \to \mathbb{R}^d$, where $\forall W,\ \forall \theta \in \Theta,\ \forall i \in \{1, d\} :\ |\theta(W)_i| \leq M$. Fitting g and θ with the procedures above using Algorithm 1, we have that with probability at least $1 - \delta$:*

$$\mathbb{E}\big[\big(A - e_0(W)\big)^T\big(\theta(W) - \theta^*(W)\big)\big]^2 \leq L_{\mathcal{D}}(\theta, g_0) - L_{\mathcal{D}}(\theta^*, g_0)$$
$$\leq \textit{Rate}_{\mathcal{D}}(S_2, \delta/2; \theta, g) + 2(1 + dM^2) \cdot \textit{Rate}_{\mathcal{D}}(S_1, \delta/2)^2.$$

Proof. Under the data model of Equations 3, 4, the categorical R-loss from Equation 5 satisfies universal Neyman orthogonality (Assumption 4, proof in Appendix B.2), and has continuous and bounded second directional derivatives (Assumption 5, proof in Appendix B.3) with $\beta = 2(1 + dM^2)$. Applying Theorem 2 from Foster & Syrgkanis (2023) concludes the proof. □

Since for the linear encoding (and for binary treatments from Proposition 2) $d = 1$, the rate of convergence for nuisance parameters is less impactful than for more flexible categorical encodings. As a result, we expect our linear encoding to be more data-efficient when the CATE at time-step t is well approximated with a linear function (the linear coefficient can still depend arbitrarily on W_t).

2.3 Causal Time Series Forecasting Models with Deep Learning

We now instantiate the theory described in §2.2 using deep-learning time-series architectures to fit the three models we require (m, e, θ). We then show how to adapt the training procedure to fit Algorithm 1. Finally, we adapt the prediction procedure to output causal forecasts.

Time-Series Architectures for Orthogonal Learning. We need out time-series architecture to take four types of inputs: static features S, temporal features $X_{1:t}$, and actions and outcome values before τ, $A_{1:\tau}$ and $Y_{1:\tau}$. We fit three models, all using the same architecture. First $e(\cdot)$ which predicts the treatment sequence $A_{>\tau}$. Second, $m(\cdot)$ which predicts expected outcomes (without knowing the treatments). Third, $\theta(\cdot)$ which predicts a vector of treatment effects (interpreted differently depending on the encoding, see §2.2). Our inputs are non-traditional, and the reason why we use a modified TFT Lim et al. (2020) (modification details in Appendix C.3). We also experiment with the state-of-the-art iTransformer architecture Liu et al. (2024), passing static features' embeddings along temporal ones, and training with randomly sampled t values by truncating the time-series at t (the iTransformer processes entire time-series and otherwise "knows the future"). This results in very slow training and prediction, and we can only apply it to one of our datasets.

Fitting a Causal Orthogonal Time-Series Model. We fit our models following Algorithm 1. We train both models $m(W_t)$, $e(W_t)$ as estimators for $\mathbb{E}[Y_t|W_t]$ and $\mathbb{E}[A_t|W_t]$ on subset S_1. $m(\cdot)$ is trained with the mean squared error (MSE) loss. The loss for $e(\cdot)$ depends on the encoding function. We use the cross entropy for the one-hot encoding case, the binary cross-entropy for the cumulative encoding, and the MSE for the linear encoding. In all cases, we train $\theta(\cdot)$ by minimizing the R-loss $l(\cdot, m, e, z_t)$ from Eq. 5 with data points $z_t \in S_2$.

Forecasting with a Causal Orthogonal Time-Series Model. θ predicts causal changes of outcomes under different treatments. However, optimizing downstream decisions (e.g. choosing prices to maximize revenue) requires a full forecast of the outcome Y_t, that changes causally with A_t. We

thus change the prediction procedure of orthogonal learning, and combine all models following Eq. 3:

$$\hat{Y}_t(W_t) = m(W_t) + \big(A_t - e(W_t)\big)^T \theta(W_t).$$

This way under g_0 we have $\hat{Y}_t(W_t) = f_0(W_t) + e_0(W_t)^T\theta(W_t) + \big(A_t - e_0(W_t)\big)^T\theta(W_t) = f_0(W_t) + A_t^T\theta(W_t)$. When $\theta(\cdot)$ is close to the CATE, our predictions change causally with A_t.

3 Regression Discontinuity Design for Causal Model Evaluation

How can we measure their performance of our causal forecasting models? This is a core challenge in causal inference, which fundamentally rely on untestable assumptions Pearl (2009), such as Assumption 1 or related data models (Eq. 1-4). In §1, we used domain knowledge on the sign of the CATE to argue that non-causal models fail to learn valid causal relationships. However, these qualitative observations do not enable quantitative comparisons between models. To the best of our knowledge, their is currently no reliable method to quantitatively evaluate the quality of causal models, without relying on the same assumptions used in fitting the models.

To address this challenge, we design another approach to estimate our causal effects of interest, that we scale to evaluate ML models. Specifically, we leverage Regression Discontinuity Designs (RDDs) (§3.1) and changes in treatments over time to estimate a subset of daily CATE values in a time-series. This approach, which uses a different assumption than Assumption 1 (§3.2), lets us identify a set of CATE values on our test set time-series, which we use to build as evaluation task for causal effects (out-of-distribution actions) on which to compare ML models (§3.3).

3.1 Background: Regression Discontinuity Designs

Regression Discontinuity Designs (RDD) leverage a continuity assumption to estimate a treatment effect at a cut-off point that triggers a change in treatment Hahn et al. (2001); Imbens & Lemieux (2008); Lee & Lemieux (2010). Formally, RDDs require a variable X with an associated cut-off value c that corresponds to a change in treatment. That is, $A = a$ when $X < c$, and $A = b$ when $X \geq c$. RDDs provide identifyability under the following assumption:

Assumption 2 (Continuity)**.** *The potential outcomes' conditional expectation,* $\mathbb{E}[Y(b)|X = x, V]$ *and* $\mathbb{E}[Y(a)|X = x, V]$ *respectively, are both continuous in* x*.*

Under assumption 2, we can estimate the CATE at the cutoff $X = c$:

Proposition 4 (CATE Identifyablity from RDD)**.** *Under Assumption 2, we have that:*

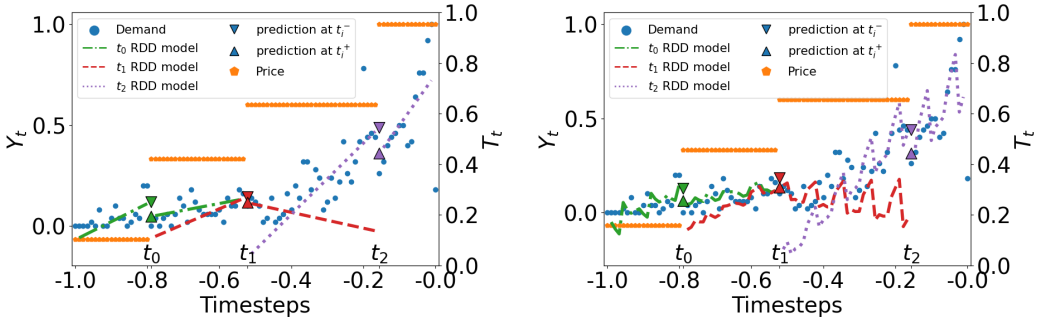
$$\mathbb{E}\big[Y(b) - Y(a) \mid X = c, V\big] = \lim_{x \to c^+} \mathbb{E}\big[Y \mid X = x, V\big] - \lim_{x \to c^-} \mathbb{E}\big[Y \mid X = x, V\big]$$

Practical estimators based on Proposition 4 typically fit linear or polynomial models $g : \mathcal{X}, \mathcal{V} \to \mathcal{Y}$, using an step variable for the cut-off $\mathbb{1}\{X = c\}$ to measure the discontinuity induced by the change of treatment. This discontinuity captures the CATE at $X = c$. Since we are interested in the limit at $X = c$, one often uses a weight kernel $K(\cdot)$ that decreases the importance of datapoints further from the cut-off $X = c$. Typical weight kernels include the rectangular "window" kernel of width h, or a triangular kernel with weights decaying linearly between $X = c$ and $(c - h, c + h)$.

3.2 Estimating Treatment Effects in Time Series

RDDs are attractive to evaluate models, as Assumption 2 differs from Assumption 1, which underpins the learning of causal models. To causally evaluate models, we want to compare a model's CATE prediction on a given time-series n at time-step t, to another estimate of this CATE used as ground truth. We obtain this ground truth by framing observed changes in treatment on individual time-series as an RDD. Consider time-series n. We call t_i^n the i^{th} *switching time-step* at which an action (price) change happens, with $a \triangleq A_{t_i^n - 1} \neq A_{t_i^n + 1} \triangleq b$. Because we use aggregated time-series (e.g., at the daily level), and treatment changes happen in the middle of an observation, $A_{t_i^n}$ is often ill defined. While this deviates from traditional RDD formulations, in which $\mathbb{E}[Y \mid X = c, V] = \mathbb{E}[Y(b) \mid X = c, V]$, Prop. 4 and associated estimators still apply.

Another deviation from §3.1 is that our time-steps are discrete, so we technically cannot take the upper- and lower-limits at the t_i^n. We follow Lee & Card (2008) and assume that specification



(a) RDDs without weekday correction

(b) RDDs using weekday correction

Figure 2: CATE estimates (distance between the two triangles) at three t_i^n values on a train time series n, without (a) and with (b) weekday correction. The RDD effectively captures the decline in demand under price increases. The correction improves fit, so CATE estimates are likely more accurate.

errors at t_i^n (the deviation from g's estimates and the true conditional expectation on each side of the cut-off) are zero in expectation. Formally:

Assumption 3 (Unbiased specification errors)**.** *Consider (continuous at c) RDD models $g_a : x \to g(a, x)$ and $g_b : x \to g(b, x)$ for time-series n. Then, $\mathbb{E}[Y_{t_i^n}(a) - g_a(c)|V] = 0$ (same for b).*

In demand forecasting, assumptions 2 and 3 with $V = \emptyset$ are unrealistic, as there are known cyclical patterns of demand based on days of the week. In §3.3, we use V to correct for these effects. With these assumptions and estimator, and denoting $w_{t_i^n}^n$ the features we give our orthogonal forecasting model at the cut-off (see §2.2), and g^n the associated RDD model, we have the following result:

Proposition 5 (RDD for point CATE)**.** *Consider time-series n and switching-time t_i^n:*

$$\mathbb{E}\big[Y_{t_i^n}(A_{t_{i+1}^n}) - Y_{t_i^n}(A_{t_i^n}) \,|W_{t_i^n} = w_{t_i^n}^n, V\big] = \lim_{t \to t_i^{n+}} \mathbb{E}\big[g^n(t) \mid V\big] - \lim_{t \to t_i^{n-}} \mathbb{E}\big[g^n(t) \mid V\big].$$

Proof. An i.i.d. time-series with context W_t is an unbiased (one point) estimate of the conditional expectation: $\mathbb{E}\big[Y_{t_i^n}(A_{t_{i+1}^n}) - Y_{t_i^n}(A_{t_i^n}) \,|W_{t_i^n} = w_{t_i^n}^n, | \; t = t_i^n\big] = \mathbb{E}\big[Y_{t_i^n}^n(A_{t_{i+1}^n}) - Y_{t_i^n}^n(A_{t_i^n})\big] = \mathbb{E}\big[Y_t^n(A_{t_{i+1}^n}) - Y_t^n(A_{t_i^n}) \mid t = t_i^n\big]$. Assumption 2 and Prop. 4 yields $\mathbb{E}\big[Y_t^n(A_{t_{i+1}^n}) - Y_t^n(A_{t_i^n}) \mid t = t_i^n\big] = \lim_{t \to t_i^{n+}} \mathbb{E}\big[Y^n \mid t, V\big] - \lim_{t \to t_i^{n-}} \mathbb{E}\big[Y^n \mid t, V\big]$. Assumption 3 gives: $\lim_{t \to t_i^{n+}} \mathbb{E}\big[Y^n \mid t, V\big] - \lim_{t \to t_i^{n-}} \mathbb{E}\big[Y^n \mid t, V\big] = \lim_{t \to t_i^{n+}} \mathbb{E}\big[g^n(t) \mid V\big] - \lim_{t \to t_i^{n-}} \mathbb{E}\big[g^n(t) \mid V\big]$. □

3.3 Causal Test Sets to Evaluate Causal Forecasting Models

Proposition 5 lets us estimate the CATE on a specific time-series n, at a specific time t_i^n, for a specific treatment change $A_{t_i^n - 1} \to A_{t_i^n + 1}$. We call this quantity $\text{CATE}_{t_i^n}$. To create our causal test set, we take the in-distribution task test set S_{test}, and collect all CATE values that we can estimate from it in $S_{\text{test}}^{\text{CATE}}$. We only include switch-times t_i^n that have at least three time-steps without treatment changes strictly before and after t_i^n, to have enough data for the RDD estimator.

Given enough data, we create a small dataset D_i^n to fit our RDD model g, including all time-steps with constant treatments around t_i^n. Formally, $D_i^n = \{t : \forall t' \in [t, t_i^n - 1],\ A_{t'} = A_{t_i^n - 1}\} \cup \{t : \forall t' \in [t_i^n + 1, t],\ A_{t'} = A_{t_i^n + 1}\}$ (note that $t_i^n \notin D_i^n$). In tasks with known cyclical patterns, such as our demand prediction task with day of the week effects (see §3.2), we create a set of comparable time-series S^n (using a rule informed by domain knowledge) on which we fit a linear regression model with time interactions $Y \sim V + V \times t$. For days of the week, this gives us parameters $\alpha_{1,j}, \alpha_{2,j}$ for $j \in \{1, \ldots, 7\}$. We then compute the residuals $\tilde{Y}_t = Y_t - \sum_{j=1}^{7}(\alpha_{1,j} + t\alpha_{2,j})$, on which we fit our RDD model. We fit L_2 regularized linear models for g, with weight kernel $K(.)$ of window size h, on dataset D_i^n using the following specification:

$$\tilde{Y}_t = \beta_0 + \beta_1(t - t_i^n) + \beta_2 \mathbb{1}_{t > t_i^n} + \beta_3 \mathbb{1}_{t > t_i^n}(t - t_i^n) + \epsilon$$

Parameter β_2 corresponds to the estimate of $\text{CATE}_{t_i^n}$ that we put in our causal test set $S_{\text{test}}^{\text{CATE}}$. Figure 2 represents the estimation of different $\text{CATE}_{t_i^n}$ for a given time series, with and without correction.

Small sample sizes in RDD datasets make the final causal test set $S_{\text{test}}^{\text{CATE}}$ noisy. We filter outliers by cutting the 2.5% tails of the distribution on each side. We evaluate models' treatment effect predictions (change of outcome when changing the action given to the model) using the RDD measurement as a ground truth, using traditional metrics such as the root-mean-square error (RMSE).

4 EVALUATION

We evaluate our approach on two tasks, and compare models on in-distribution forecasts and our RDD-based causal task that evaluates a model's ability to predict on out-of-distribution actions. Causal models perform worse in distribution as they do not "cheat" by using the confounded choice of actions as signal. As a result, their predictions are better on out-of-distribution actions.

4.1 DEMAND FORECASTING DATASET

Dataset. Each time-series is a sequence of prices and number of sales for one train. Temporal features include the weekday of the timestep and the number of days before departure. Stationary features include the departure and arrival terminals and departure date time. The dataset consists in about 300,000 training time-series, and validation and test sets of around 100,000 each. All the figures and metrics regarding this dataset have been anonymized. Time steps t are normalized to $[-1, 0]$ (all time series have the same length).

Experimental setup. We set $\tau = 0.33$, and forecast all outcomes $(Y_t)_{t>\tau}$. Baseline (non-causal) models are trained with the MSE loss, and include a well-tuned LGBM model (used in production) and two deep learning architectures: the state-of-the-art iTransformer Liu et al. (2024), and a custom version of the Temporal Fusion Transformer (TFT) Lim et al. (2020). The TFT maps well to our requirements (see §2), and we modify it for performance (details in Appendix C.3). We tune all models' hyper-parameters separately and equally, and report metrics by training and testing with 5 random seeds on final hyper-parameters, and reporting the mean and standard-deviation. The RDD dataset uses a linear weights kernel with $h = 14$ time steps. Keeping all switching times with at least 3 data points strictly before and after the price change retains 70% of observed price changes.

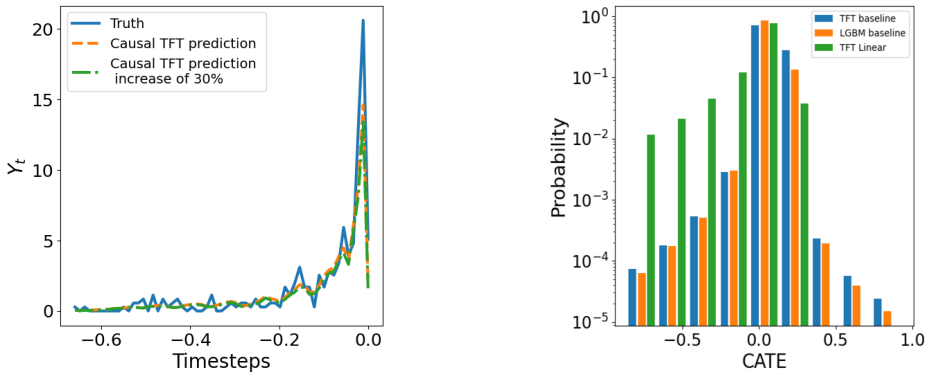
Results. Table 1 shows the performance of several models. Non-causal models perform best in-distribution (RMSE), and the LGBM is best among them. This is expected: the observed prices are very informative in-distribution, since operators use their experience to set prices based on the demand they expect. On causal tasks however (predicting the effect of a price change), causal models outperform the baselines (RDD RMSE). The linear encoding is particularly accurate, which we attribute to the high treatment dimension d (§2.2). The Causal iTransformer with linear encoding outperforms the baseline TFT (best non-causal model) by 37%, a significant improvement. Figure 3a shows the causal TFT on the same time series as figure 1b-1a: the causal model captures the correct CATE sign (increasing prices decreases the demand forecast). Figure 3b confirms that the causal TFT's CATE distribution is qualitatively better. On the test set, the causal TFT predicts fewer positive CATE effects than baselines. The positive CATEs are also smaller.

Models	RMSE	RDD RMSE
LGBM	**0.6870 ± 0.0015**	0.2163 ± 0.0009
TFT Baseline	0.7044 ± 0.0050	0.1910 ± 0.0014
Causal TFT Linear	0.7216 ± 0.0020	**0.1217 ± 0.0019**
Causal TFT Cumulative	0.7189 ± 0.0091	0.1820 ± 0.0008
Causal TFT One-hot	0.7236 ± 0.0115	0.1874 ± 0.0007
Causal iTransformer Linear	0.7504 ± 0.0098	**0.1197 ± 0.0014**
Causal iTransformer Cumulative	0.7377 ± 0.0070	0.1908 ± 0.0021
Causal iTransformer One-hot	0.7347 ± 0.0097	0.1930 ± 0.0023

Table 1: Baselines are good in distribution (RMSE), but causal models capture causal effects of actions changes (RDD RMSE).

4.2 MIMIC HEALTH PREDICTION DATASET

Dataset. To evaluate our approach on a public tsk, we use MIMIC-extract library Wang et al. (2020) to process the MIMIC-III Johnson et al. (2016) heatlh dataset. Every time-series represents data from one patient who stayed between 30 and 60 hours in a critical care unit, a typical setting in causal inference benchmarks Bica et al. (2020a); Lim (2018); Melnychuk et al. (2022). Static features include the patient's gender, age, and ethnicity. Temporal features include 25 vital signals. Our task is to forecast a patient's blood pressure, and we aim at estimating the treatment effect of



(a) Causal TFT

(b) Treatment effects distribution

Figure 3: (a) Daily demand forecast and its change under price increase for one train, causal TFT. Analogous to Fig. 1b-1a. (b): CATE distribution over the test set, for a change from the observed price to the next possible price, normalized by the price change. Analogous to Fig. 1c.

Models	$t = \tau$+1	$t = \tau$+2	$t = \tau$+3	$t = \tau$+4	$t = \tau$+5
In distribution RMSE					
Causal Transformer	**8.823 ± 0.028**	**9.499 ± 0.024**	9.845 ± 0.035	10.098 ± 0.028	10.313 ± 0.026
TFT Baseline	8.921 ± 0.056	9.534 ± 0.057	**9.834 ± 0.061**	**10.051 ± 0.052**	**10.238 ± 0.043**
Causal TFT One-hot	8.951 ± 0.023	9.596 ± 0.025	9.921 ± 0.038	10.165 ± 0.028	10.371 ± 0.027
Causal TFT Cumulative	8.945 ± 0.026	9.608 ± 0.033	9.935 ± 0.046	10.184 ± 0.035	10.390 ± 0.030
RDD RMSE for CATE prediction					
Causal Transformer	2.887 ± 0.145	2.869 ± 0.145	**2.844 ± 0.141**	**2.833 ± 0.140**	**2.833 ± 0.129**
TFT Baseline	2.908 ± 0.102	2.942 ± 0.103	2.969 ± 0.114	3.011 ± 0.104	3.051 ± 0.096
Causal TFT One-hot	**2.861 ± 0.094**	**2.861 ± 0.094**	2.861 ± 0.094	2.862 ± 0.094	2.862 ± 0.094
Causal TFT Cumulative	2.895 ± 0.050	2.895 ± 0.050	2.895 ± 0.050	2.895 ± 0.050	2.895 ± 0.050

Table 2: Performance of next five steps forecasts, in distribution and for causal effects (RDD).

mechanical ventilation and vasopressor drugs. Since this represents a categorical treatment, our linear encoding does not apply.

Experimental setup. We keep the default variable length time-series, and follow prior art in forecasting blood pressure for the next 5 hours (Lim, 2018; Melnychuk et al., 2022). That is, for each t we set $\tau = t - 1$ and forecast $[t, t + 4]$. We use the same baseline losses as in §4.1. We were not able to fit our causal models with the iTransformer backbone, which requires a fixed τ, so we focus on our causal TFT. We also consider a state-of-the-art causal forecasting baseline, the Causal Transformer Melnychuk et al. (2022), modified to remove information leakage from the future vitals of hour $t = \tau + 1$ (details in §C.1). We evaluate forecasts in distribution as prior work, as well as using our RDD methodology. Our RDDs use a rectangular weight kernel with $h = 5$. Requiring at least three points strictly on each side preserves about 70% of the switching times.

Results. We show RMSE results, D.2 shows similar results for the MAE. Table 2 (up) shows that baseline models outperform causal ones in distribution. The auto-regressive Causal Transformer outperforms our baseline TFT in next-step prediction, but is less accurate further into the future, which is consistent with prior observations Liu et al. (2024); Das et al. (2024); Wu et al. (2021); Ilbert et al. (2024). Once again, baseline models are able to exploit the correlation between treatments and future outcomes, as doctors use their domain knowledge to anticipate and prevent issues.

On the causal task (out-of-distribution actions) using RDD-based CATE estimates, Table 2 (down) shows that the one-hot causal model outperforms the baseline TFT by at least 1.6% in all time-steps, and the Causal Transformer by almost 1% in the first time step. Contrary to our Causal TFT, the Causal Transformer has access to the treatment sequence between τ and t. Since our RDD methodology only estimates daily treatment effects, the sequence of treatments $\tau : t$ is always in distribution, and the Causal Transformer is able to exploit this signal. However, this means that Causal Transformer CATE predictions for changes in *treatment sequences* would be confounded, and suffer some the same issues as the TFT baseline. Finding similar estimators to our RDD approach for treatment sequences to rigorously evaluate these effects is an interesting avenue for future work.

REFERENCES

Ioana Bica, Ahmed M. Alaa, James Jordon, and Mihaela van der Schaar. Estimating counterfactual treatment outcomes over time through adversarially balanced representations, 2020a. URL https://arxiv.org/abs/2002.04083.

Ioana Bica, Ahmed M. Alaa, and Mihaela van der Schaar. Time series deconfounder: Estimating treatment effects over time in the presence of hidden confounders, 2020b. URL https://arxiv.org/abs/1902.00450.

Kay H. Brodersen, Fabian Gallusser, Jim Koehler, Nicolas Remy, and Steven L. Scott. Inferring causal impact using bayesian structural time-series models. *The Annals of Applied Statistics*, 2015. URL http://dx.doi.org/10.1214/14-AOAS788.

Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O. Arik, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecasting, 2023. URL https://arxiv.org/abs/2303.06053.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *Econometrics Journal*, 2018.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder, 2024. URL https://arxiv.org/abs/2304.08424.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. URL https://arxiv.org/abs/2105.05233.

Dylan J Foster and Vasilis Syrgkanis. Orthogonal statistical learning. *The Annals of Statistics*, 2023.

Jinyong Hahn, Petra Todd, and Wilbert Van der Klaauw. Identification and estimation of treatment effects with a regression-discontinuity design. *Econometrica*, 2001.

Catherine Hausman and David S Rapson. Regression discontinuity in time: Considerations for empirical applications. *Annual Review of Economics*, 2018.

Paul W Holland. Statistics and causal inference. *Journal of the American statistical Association*, 1986.

Romain Ilbert, Ambroise Odonnat, Vasilii Feofanov, Aladin Virmaux, Giuseppe Paolo, Themis Palpanas, and Ievgen Redko. Samformer: Unlocking the potential of transformers in time series forecasting with sharpness-aware minimization and channel-wise attention, 2024. URL https://arxiv.org/abs/2402.10198.

Guido W Imbens and Thomas Lemieux. Regression discontinuity designs: A guide to practice. *Journal of econometrics*, 2008.

Guido W. Imbens and Donald B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press, 2015.

Ziwei Ji, Miroslav Dudík, Robert E Schapire, and Matus Telgarsky. Gradient descent follows the regularization path for general losses. In *Conference on Learning Theory*, 2020.

Alistair Johnson, Tom Pollard, Lu Shen, Li-wei Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Celi, and Roger Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3:160035, 05 2016. doi: 10.1038/sdata.2016.35.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

David S Lee and David Card. Regression discontinuity inference with specification error. *Journal of Econometrics*, 2008.

David S Lee and Thomas Lemieux. Regression discontinuity designs in economics. *Journal of economic literature*, 2010.

Rui Li, Zach Shahn, Jun Li, Mingyu Lu, Prithwish Chakraborty, Daby Sow, Mohamed Ghalwash, and Li wei H. Lehman. G-net: A deep learning approach to g-computation for counterfactual outcome prediction under dynamic treatment regimes, 2020a. URL https://arxiv.org/abs/2003.10551.

Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting, 2020b. URL https://arxiv.org/abs/1907.00235.

Bryan Lim. Forecasting treatment responses over time using recurrent marginal structural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/56e6a93212e4482d99c84a639d254b67-Paper.pdf.

Bryan Lim, Sercan O. Arik, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting, 2020. URL https://arxiv.org/abs/1912.09363.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting, 2024. URL https://arxiv.org/abs/2310.06625.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 2022. URL https://www.sciencedirect.com/science/article/pii/S0169207021001874.

Spyros Makridakis, Evangelos Spiliotis, Ross Hollyman, Fotios Petropoulos, Norman Swanson, and Anil Gaba. The m6 forecasting competition: Bridging the gap between forecasting and investment decisions, 2023. URL https://arxiv.org/abs/2310.13357.

Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Causal transformer for estimating counterfactual outcomes, 2022. URL https://arxiv.org/abs/2204.07258.

X Nie and S Wager. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108 (2):299–319, 09 2020. ISSN 0006-3444. doi: 10.1093/biomet/asaa076. URL https://doi.org/10.1093/biomet/asaa076.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers, 2023. URL https://arxiv.org/abs/2211.14730.

Zhale Nowroozilarki, Arash Pakbin, James Royalty, Donald K.K. Lee, and Bobak J. Mortazavi. Real-time mortality prediction using mimic-iv icu data via boosted nonparametric hazards. IEEE, 2021. URL http://dx.doi.org/10.1109/BHI50953.2021.9508537.

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

P. M. Robinson. Root-n-consistent semiparametric regression. *Econometrica*, 1988.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Douglas Schultz, Johannes Stephan, Julian Sieber, Trudie Yeh, Manuel Kunz, Patrick Doupe, and Tim Januschowski. Causal forecasting for pricing, 2024. URL https://arxiv.org/abs/2312.15282.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 2018.

Shirly Wang, Matthew B. A. 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*, CHIL '20, pp. 222–235. Association for Computing Machinery, 2020. doi: 10.1145/3368555.3384469. URL https://doi.org/10.1145/3368555.3384469.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, 2021.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. 2023. URL https://openreview.net/forum?id=vSVLM2j9eie.

A Encoding functions

In this section, we expand on the three different encoding functions that we plug into the R-Loss.

A.1 Linear encoding

A possible encoding function is to pass the treatment as a scalar value in the R-Loss, giving a linear approximation of $\mathbb{E}[Y_t(a)|W_t]$ with regards to a. In this framework, we denote A_t as the scalar value of the treatment. Hence the data model reduces to that of Eq. 1, but with a continuous a.

We can access the CATE between a and b as:

$$\mathbb{E}[Y(A_t = b, W_t) - Y(A_t = a, W_t)|W_t] = \theta_0(W_t)(b - a)$$

This encoding performs best on the the railway dataset, in which treatments are fairly high dimensional and, conditioned on W_t, we only observe local variation of the treatment around a typical price. Figure 4 is an example of the estimated values of θ_0 for a random time-series. We notice that the effect of prices has little impact at the beginning of the forecast. The effect of price over sales increases as the departure date gets closer. Moreover, we observe that the price has little effect at time step $t = 0$ (the day at which the train leaves leaves).

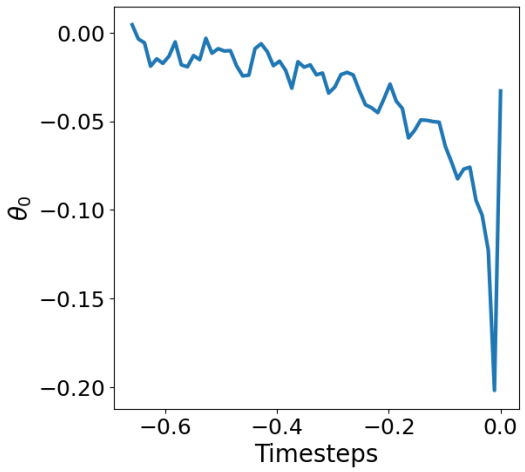


Figure 4: θ_0 values for various time steps for a railway time serie

A.2 ONE-HOT ENCODING

A more intuitive encoding under our categorical treatments is the one-hot encoding. In this case $e_0(W_t)_i = P(A_t = i|W_t)$. With d possible treatments, the vector $\theta_0(W_t)$ is:

$$\forall i \in [1:k], \theta_0(W_t)_i = \mathbb{E}[Y_t(A_t = i, W_t) - f_0(W_t)|W_t] \tag{6}$$

In order to estimate the CATE between a and b, we compute the difference between $\theta_0(W_t)_b$ and $\theta_0(W_t)_a$:

$$\mathbb{E}[Y(A_t = b, W_t) - Y(A_t = a, W_t)|W_t] = \theta_0(W_t)_b - \theta_0(W_t)_a$$

Thanks to equation 6 and to domain knowledge on the railway dataset, we can make the assumption that the values of $\theta_0(W_t)_i$ must decrease when i increases. Figure 5 shows an example of θ's vector values for a specific time-series and time step. We notice that the values seem to decrease as the treatment increases. Nonetheless, we observe an increase for small treatment values. It is likely caused by the fact that those treatments are extremely rare in our dataset (conditioned on W_t).

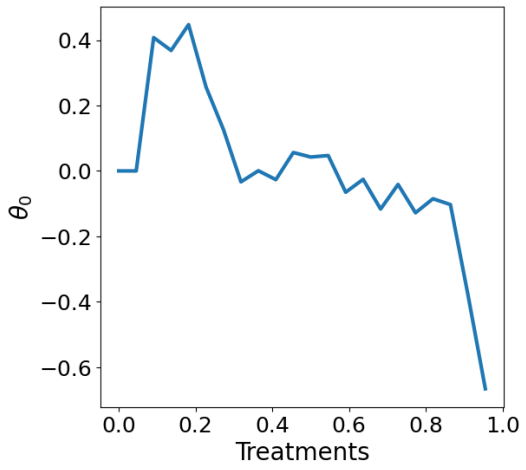


Figure 5: Example of θ vector values for the one-hot encoding on the passenger rail dataset.

A.3 CUMULATIVE ENCODING

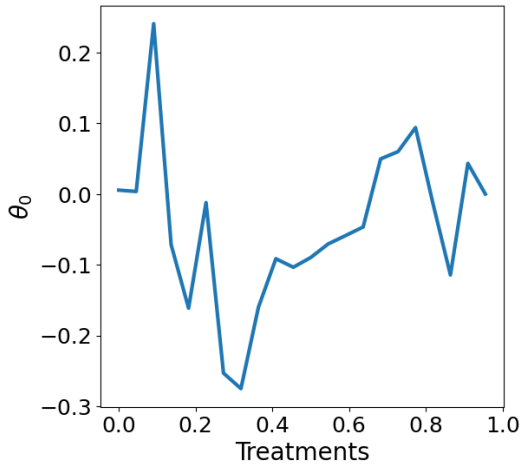


Figure 6: Example of θ_0 vector values for the cumulative encoding

The cumulative encoding function $x \mapsto \mathbb{1}_{A_t \geq x}$ is another possible encoding function of the treatment in our data model in Eq. 3. The interpretation of the vector $\theta_0(W_t)$ is then:

$$\forall i \in [2:k], \theta_0(W_t)_i = \mathbb{E}[Y_t(A_t = i, W_t) - Y_t(A_t = i-1, W_t)|W_t]$$

Thus the CATE between treatment a and b, with $a < b$ equals:

$$\mathbb{E}[Y(A_t = b, W_t) - Y(A_t = a, W_t)|W_t] = \sum_{i=a+1}^{b} \theta_0(W_t)_i$$

As done in the last subsection, we also can infer some characteristics of the theta vector values thanks to domain knowledge and equation A.3. In that case, the values of θ_0 should be negative for all indexes. Figure 6 shows an example of $\theta(W_t)$. Most of them are indeed negative. Nonetheless, some values associated to small treatment values, and a few very large treatment values, are still positive. Again, this is likely caused by those treatments being rarely seen during training (conditioned on W_t).

B Detailed Theory and Proofs

B.1 Orthogonal Learning Assumptions

Proposition 3 relies on two assumptions that must be verified by the R-loss to ensure convergence through Theorem 2 in Foster & Syrgkanis (2023). These two assumptions are **Universal Orthogonality** and **Boundness**. Denote the observations $Z^n = (Y^n_{1:t}, A^n_{1:t}, X^n_{1:t}, S^n)$, sampled from an unknown distribution $\mathcal{D}_t$ depending on t. We denote $\theta^* \in \arg\min_{\theta\in\Theta} L_{\mathcal{D}_t}(\theta, m, e)$, and $star(\Theta, \theta)$ the star domain of Θ at point θ. Both assumptions require the definition of a *Directional Derivative*:

Definition 1 (Directional Derivative)**.** *Let $F : \mathcal{F} \to \mathbb{R}$ be a function from a vector space $\mathcal{F}$, and define the derivative operator at point f with respect to vector g as:*

$$D_f F(f)[g] = \frac{d}{dt} F(f + tg)_{|t=0}$$

With this definition, we can state the two assumptions required of the (population level) R-loss (we omit the indexing on t in $\mathcal{D}_t$, as the same assumption must old in each t but our definition is identical at each t):

Assumption 4 (Universal Orthogonality)**.** *For all $\hat{\theta} \in star(\hat{\Theta}, \theta^*) + star(\hat{\Theta} - \theta^*)$:*

$$D_g D_\theta L_D(\hat{\theta}, m, e)[\theta - \theta^*, g - g_0] = 0$$

Assumption 5 (Boundness)**.** *$D_g D_g L_D$ and $D_\theta D_\theta D_g L_D$ are both continuous and there is a constant β such as $\forall\theta \in star(\hat{\Theta}, \theta^*)$ and $\forall\tilde{g} \in star(\mathcal{G}, g_0)$:*

$$|D_g D_g L_D(\theta, \tilde{g})[g - g_0, g - g_0]| \leq \beta||g - g_0||^2_{\mathcal{G}}$$

B.2 Universal Orthogonality

We denote $W_t = (S, X_{1:t}, Y_{1:\tau})$. We omit the indices t for A_t, Y_t and W_t. We note $g_0 = (m_0, e_0)$. Next, we prove the orthogonality of the R-Loss.

$$\begin{aligned}
&D_\theta L_D(\bar{\theta}, g_0)[\theta - \theta^\star] \\
&= -2\mathbb{E}[(A - e_0(W)^T\big(\theta(W) - \theta^\star(W)\big) \times \big(Y - m_0(W) - (A - e_0(W))^T\bar{\theta}(W)\big)]
\end{aligned}$$

We first consider the directional derive with respect to e:

$$\begin{aligned}
&D_e D_\theta L_D(\bar{\theta}, g_0)[\theta - \theta^\star, e - e_0] = \\
&2\mathbb{E}[\big(Y - m_0(W) - (A - e_0(W))^T\bar{\theta}(W)\big) \times \big(e(W) - e_0(W)\big)^T\big(\theta(W) - \theta^\star(W)\big)] \\
&- 2\mathbb{E}[\big(e(W) - e_0(W)\big)^T\bar{\theta}(W) \times \big(A - e_0(W)\big)^T\big(\theta(W) - \theta^\star(W)\big)]
\end{aligned}$$

We notice that in the first term of this derivative, we have $\forall w \in \mathcal{W}$:

$$\mathbb{E}[((Y - m_0(W) - (A - e_0(W))^T\bar{\theta}(W)|W = w] = \mathbb{E}[\epsilon_1 + \epsilon_2^T\big(\theta_0(W) - \bar{\theta}(W)\big)|W = w] = 0$$

Regarding the second term, when developing the dot products we also have that $\forall w \in \mathcal{W}$:

$$\begin{aligned}
&\mathbb{E}[(e(W) - e_0(W))^T \bar{\theta}(W) \times (A - e_0(W))^T (\theta(W) - \theta^\star(W)) | W = w] \\
&= \sum_{i=1}^{k} \sum_{j=1}^{k} \mathbb{E}[\epsilon_{2j} | W = w] \times \bar{\theta}(w)_j \times (e(w) - e_0(w))_i \times (\theta(w) - \theta_0(w))_j \\
&= 0
\end{aligned}$$

We now need to establish the orthogonality for the parameter m of the population risk:

$$\begin{aligned}
&D_m D_\theta L_D(\bar{\theta}, g_0)[\theta - \theta^\star, m - m_0] \\
&= 2\mathbb{E}[(A - e_0(W))^T (\theta(W) - \theta^\star(W)) \times (m(W) - m_0(W))] \\
&= 2\mathbb{E}[\epsilon_2^T (\theta(W) - \theta^\star(W)) \times (m(W) - m_0(W))] \\
&= 0
\end{aligned}$$

with $E[\epsilon_2 | Z] = 0_d$.

Hence, the population level version of the R-loss from Eq. 5 verifies assumption 4.

B.3 Boundness

The second directional derivatives of the population risk are continuous as it is the composition the square function and a linear function, both with continuous gradients. We now prove the boundedness of the second derivative of the population risk with regard to the nuisance parameters, for the vector case.

Let us denote $l(\gamma, \zeta, z) = [Y - \gamma_1 - (A - \gamma_2)^T \zeta]^2$ with $\gamma_1 \in \mathbb{R}$ and $(\gamma_2, \zeta) \in (\mathbb{R}^{\mathrm{d}})^2$ We have:

$$\nabla_{\gamma\gamma} l(\gamma, \zeta, z) = 2 \left[\begin{array}{c|c} 1 & -\zeta_1 \cdots\cdots -\zeta_d \\ \hline -\zeta_1 & \\ \vdots & (\zeta_i \zeta_j)_{(i,j) \in [1,d]^2} \\ \vdots & \\ -\zeta_d & \end{array}\right]$$

This matrix rank is 1 as every line i is equal to the first line multiplied by $-\zeta_i$. Moreover the unique non-zero eigenvalue of this matrix is $2 * (1 + \sum_{i=1}^{d} \zeta_i^2)$. Its eigenvector is $(1, -\zeta_1, ..., -\zeta_d)$.

Denote M the bound such as $\forall W, \forall i, |\theta(W)_i| < M$. Then when $\zeta_i = \theta(W)_i$, we have that $2 * (1 + \sum_{i=1}^{d} \zeta_i^2) \leq 2(1 + dM^2)$. The directional derivative of the R-loss function in Eq. 5 at $g - g_0$ is the derivative of the composition of the functions 5 and $\tilde{g} + t(g - g_0)$, with respect to t. Hence, when taking two derivatives with respect to t, the equation become:

$$||D_g D_g R(\theta, \tilde{g}, z)[g - g_0, g - g_0]|| = ||(g - g_0)^T \nabla_{\gamma\gamma} l(\gamma, \zeta, z)(g - g_0)||$$

as the hessian of $t \mapsto t(g - g_0)$ equals 0.

Using the fact that the largest eigenvalue of $\nabla_{\gamma\gamma} l(\gamma, \zeta, z)$ is less than $2(1 + dM^2)$, we have $||D_g D_g R(\theta, \tilde{g})[g - g_0, g - g_0]|| < 2(1 + dM^2)||g - g_0||_{\mathcal{G}}^2$. Finally, we compute the expectation on this inequality and we apply the dominated convergence theorem to get the population risk (requiring that $||g - g_0||_{L_2} < \infty$). To conclude, we get the following inequality:

$$||D_g D_g L_D(\theta, \tilde{g})[g - g_0, g - g_0]|| < 2(1 + d \times M^2)||g - g_0||_{L_2}^2$$

Thus, assumption 5 is verified with $\beta = 2(1 + dM^2)$.

C Implementation details

C.1 Causal Transformer

We use the Causal Transformer (Melnychuk et al., 2022) in order to get a reference model for comparison. We use the public github repository (`https://github.com/Valentyn1997/`

CausalTransformer/tree/main). Melnychuk et al. (2022) train an auto regressive model, like us using 25 vital signals, the age, the gender and the ethnicity to estimate the blood pressure of patient. In that context, the relevant features for predicting blood pressure at time t are all the vitals before time t, static features and all treatments up until time t. Nonetheless, when evaluating for time $\tau + l$ for l in $[1, 5]$, the model had knowledge of vitals at time $\tau + 1$, enabling the model to perform very well for $l = 1$ and leaking information for all τ (https://github.com/Valentyn1997/CausalTransformer/blob/c49a24faa57af966501e241f57a26b528f874a53/src/data/mimic_iii/real_dataset.py#L67). This leak manifested as extremely good performance for $l = 1$, followed by a large drop of performances at $l \geq 2$. In our evaluation, we fix the leak from the future, and obtain results consistent with other models trained on this dataset during our study.

C.2 RDD

Algorithm 2 describes how to compute the dataset of RDD values in full details.

Algorithm 2 RDD Algorithm

Input: N time series (Y_t, A_t, W_t)

Initialize $S_{\text{test}}^{\text{CATE}} \leftarrow \{\}$

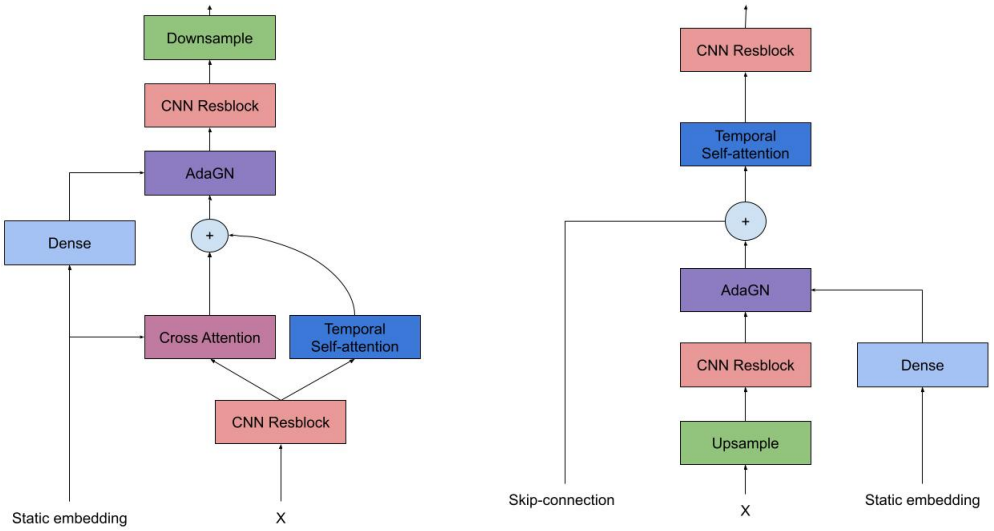
for n **from** 1 **to** N **do**
  Initialize $k \leftarrow 1$
  for t **from** 1 **to** length$(Y^n) - 1$ **do**
    if $T_t \neq T_{t+1}$ **then**
      $t_k^n \leftarrow t$
      $k \leftarrow k + 1$
    end if
  end for
  if $k > 2$ **then**
    for i **from** 1 **to** $k - 1$ **do**
      if $(t_i^n - t_{i+1}^n) \geq 2$ **and** $((i = 1$ **and** $t_i^n \geq 1)$ **or** $(t_i^n - t_{i-1}^n) \geq 2)$ **then**
        $D_i^n \leftarrow \left\{t_{i-1}^n + 1, t_{i-1}^n + 2, ..., t_{i+1}^n - 1\right\} \setminus \{t_i^n\}$
        for t **in** D_i^n **do**
          $\tilde{X}_t \leftarrow \left(t - t_i^n, \mathbb{1}_{\mathrm{t>t_i^n}}, \mathbb{1}_{\mathrm{t>t_i^n}} \times (t - t_i^n)\right)$
        end for
        $f \leftarrow \text{LinearModel}.fit((\tilde{X}_t)_{t \in D_i^n}, (Y_t)_{t \in D_i^n}, (K(t))_{t \in D_i^n})$
        Append f.coefficients[2] to $S_{\text{test}}^{\text{CATE}}$ ▷ Extract β_2 from the model f
      end if
    end for
  end if
end for
Return $S_{\text{test}}^{\text{CATE}}$

C.3 TFT

We developed a custom version of the Temporal Fusion Transformer Lim et al. (2020). We focused on this architecture as it supports static features, past and futures features as input, and temporally causal processing of the input, where most architectures focus on predicting future time step from the previous ones Liu et al. (2024),Zhang & Yan (2023),Wu et al. (2021),Nie et al. (2023). We apply several changes to the original TFT:

- We remove the encoder of the TFT and replace it by a series of encoder blocks. Figure 7a presents the architecture of our encoder block. Each encoder block is made of self-attention and residual blocks. Each block divides the length of the input by 2. Hence, at the beginning of the encoder, the model processes high frequency features, and the deeper the encoder is the lower frequency the model is considering.

- We use conditional mechanisms that show interesting performances on Image Generation. Static features go through a dense model to create an embedding s gathering all the static information. We then apply successive Cross Attention mechanisms between the token s and the tokens of the temporal features Rombach et al. (2022) to mix temporal and static features. Finally, we apply adaptive group normalization Dhariwal & Nichol (2021) where s is the class embedding.
- We replace the masked self-attention decoder by a new decoder block. Figure 7b shows the layers used in the decoder block. This block is almost the symmetric of the encoder one. We remove the cross-attention layer in the decoder as it increases the expressivity of the model. We also add skip connections between the encoder the decoder blocks as done in Ronneberger et al. (2015). Thus, the information of high frequency features are directly transferred to the decoder without passing thought the entire encoder in order to help the expressivity of the model.
- We use causal convolution layers to prevent any possible leak of information from future features X_{t+i} to impact Y_t prediction. We also apply masking in our self-attention layers as done in Li et al. (2020b).



(a) Encoder block (b) Decoder block

Figure 7: Our modified TFT architecture.

D Additional results

D.1 Addition demand forecasting results

Table 3 shows MAE results for in-distribution forecasts and our RDD dataset. MAE metrics show less variation overall, though qualitative results are similar to those described in §4.1 for the RMSE and RDD RMSE.

D.2 Additional MIMIC-II results

Table 4 shows MAE results for in-distribution forecasts. Table 5 shows MAE results for causal effects using our RDD dataset. Results show the same effects as those described in §4.2 for the RMSE and RDD RMSE.

Models	RMSE	MAE	RDD RMSE	RDD MAE
LGBM	**0.6870 ± 0.0015**	**0.2877 ± 0.0005**	0.2163 ± 0.0009	0.1390 ± 0.0004
TFT Baseline	0.7044 ± 0.0050	0.3023 ± 0.0053	0.1910 ± 0.0014	0.1262 ± 0.0005
Causal TFT Linear	0.7216 ± 0.0020	0.3104 ± 0.0047	**0.1217 ± 0.0019**	0.1271 ± 0.0008
Causal TFT Cumulative	0.7189 ± 0.0091	0.3067 ± 0.0019	0.1820 ± 0.0008	**0.1243 ± 0.0004**
Causal TFT One-hot	0.7236 ± 0.0115	0.3095 ± 0.0069	0.1874 ± 0.0007	0.1279 ± 0.0002
Causal iTransformer Linear	0.7504 ± 0.0098	0.3113 ± 0.0038	**0.1197 ± 0.0014**	0.1259 ± 0.0003
Causal iTransformer Cumulative	0.7377 ± 0.0070	0.3058 ± 0.0017	0.1908 ± 0.0021	0.1282 ± 0.0009
Causal iTransformer One-hot	0.7347 ± 0.0097	0.3036 ± 0.0030	0.1930 ± 0.0023	0.1288 ± 0.0009

Table 3: Baselines perform better in distribution (RMSE, MAE), while causal models better capture causal effects (RDD metrics).

Models	$t = \tau$+1	$t = \tau$+2	$t = \tau$+3	$t = \tau$+4	$t = \tau$+5
Causal Transformer	**6.028 ± 0.027**	**6.703 ± 0.027**	7.048 ± 0.025	7.290 ± 0.032	7.485 ± 0.040
TFT Baseline	6.162 ± 0.044	6.745 ± 0.024	**7.032 ± 0.018**	**7.233 ± 0.022**	**7.402 ± 0.024**
Causal TFT One-hot	6.170 ± 0.030	6.802 ± 0.033	7.115 ± 0.033	7.337 ± 0.036	7.515 ± 0.033
Causal TFT Cumulative	6.147 ± 0.045	6.799 ± 0.054	7.125 ± 0.044	7.359 ± 0.05	7.544 ± 0.051

Table 4: MAE with respect to the ground truth time series, best values are in **bold**

Models	$t = \tau$+1	$t = \tau$+2	$t = \tau$+3	$t = \tau$+4	$t = \tau$+5
Causal Transformer	2.286 ± 0.096	2.271 ± 0.097	**2.249 ± 0.096**	**2.241 ± 0.094**	**2.242 ± 0.085**
TFT Baseline	2.301 ± 0.065	2.320 ± 0.067	2.337 ± 0.071	2.368 ± 0.065	2.400 ± 0.058
Causal TFT One-hot	**2.265 ± 0.066**	**2.265 ± 0.066**	2.265 ± 0.066	2.265 ± 0.066	2.265 ± 0.066
Causal TFT Cumulative	2.292 ± 0.044	2.292 ± 0.044	2.292 ± 0.044	2.292 ± 0.044	2.292 ± 0.044

Table 5: RDD MAE w.r.t the CATE, best values are in **bold**

E Extended Related Work

Our two background sections §2.1 and §3.1 already discuss the closest related work. In this section, we expand on this discussion and discuss other related papers.

Time-series forecasting has seen a flurry of recent activity. The main task studied in the literature is that of multivariate time series forecasting, with the goal of forecasting $(Y_i^n)_{i\in[t:t+h]}$ from $(Y_i^n)_{i\in[0:t]}$. Another common task is to forecast $(Y_i^n)_{i\in[1:t]}$ from $(X_i^n)_{i\in[1:t]}$. Various approaches have been proposed, though not clear winner has emerged yet. TSMixer Chen et al. (2023) relies only on fully connected linear layers to process entire time-series feature-wise, before mixing features at each time-step. The iTransformer Liu et al. (2024) architecture similarly leverages fully connected layers, but processes intermediate representations with transformer layers channel wise. PatchTST Nie et al. (2023) gathers patches time-steps and features that it treats as tokens in a self-attention mechanism, before reconstructing a time-series shaped output. In all these architectures and others Ilbert et al. (2024); Wu et al. (2021), there is no insurance against future information leaking into previous time-steps. While this can be fine in some cases, this is not compatible with our specific task. Indeed, our causal models predict Y_{t+l}^n based on past features $(Y_i^n)_{i\in[0:t]}$, static features S^n and temporal features $(X_i^n)_{i\in[0:t+l]}$. A key point in our task is to ensure causality in the forecasting, preventing any leak of information from $(X_i^n)_{i\in[t+1:t+h]}$. Supporting this requirement is not common in recent time series papers, and the reason why we build on the TFT Lim et al. (2020), which does provide an adequate structure.

Causal time-series models have seen less progress, though some time-series models aim to estimate the CATE. Bica et al. (2020b); Lim (2018) rely on the same meta-architecture, including a network computing the treatment propensity score and a network computing the outcome (though

Bica et al. (2020b) tries to learn unobserved confounders, a challenging task). Both models use LSTM or RNN architectures merged with new mechanisms: a propensity network learns to predict propensity scores, which are then used to train the outcome network with the inverse probability of treatment weighting. This loss, mainly used for policy learning Foster & Syrgkanis (2023), can be high variance and does not full corrects for regularization bias. This is why we decided to train our model with the Residualized Loss (R-Loss) Foster & Syrgkanis (2023); Nie & Wager (2020), which also gives us a more flexible approach for treatment encoding.

Brodersen et al. (2015) take a different approach, and use structured time-series models to explicitly model counterfactuals under no-treatment as a synthetic control. This approach is well suited to binary treatments with several observations, to be able to model outcomes without treatment. It is less applicable to our case with high dimensional treatments and complex conditioning variables. However, it could interesting to explore this approach as a multi-time-steps alternative to our RDD causal test set.

With the recent rise of the transformer architecture for sequential models, new techniques have been developed to merge transformers with causal inference Melnychuk et al. (2022). The authors leverage an adversarial loss Bica et al. (2020a) to encourage the model to learn balanced features for the treated and untreated populations and learn the causal link between the treatment and the outcome. Melnychuk et al. (2022) use a transformer backbone with two small dense layers on top trained with this adversarial loss. This identification approach differs from classical causal inference frameworks, and it is unclear that it converges to a well defined conditional causal effect.

Regression Discontinuity Designs (RDDs) are studied theoretically, and frequently used empirically, in economics Hahn et al. (2001); Imbens & Lemieux (2008); Lee & Lemieux (2010). Researchers typically use RDDs to estimate one specific ATE. In contrast, we estimate many ATEs (one for each time series), and combine them with time-series features W_t to create a CATE dataset. To the best of our knowledge, this is a novel appraoch to evaluate causal models. Researchers have used RDDs to estimate ATEs in time-series before though. Hausman & Rapson (2018) surveys typical approaches and pitfalls, and proposes empirical checks to perform on the data to verify if RDDs are applicable. In our setting, we typically do not have enough data around switching times to perform such checks, though it would be interesting to take inspiration from common RDD checks to improve our causal test sets. We leave this direction for future work.