Submitted to *NBER Summer Institute*

FORECASTING SYNTHETIC CONTROL

PRISCILA GRECOV
Department of Data Science and Artificial Intelligence, Monash University

KLAUS ACKERMANN
Department of Econometrics and Business Statistics, Monash University

CHRISTOPH BERGMEIR
Department of Computer Science and Artificial Intelligence, University of Granada and Department of Data Science and Artificial Intelligence, Monash University

We introduce a novel forecasting method employing global deep learning models for estimating the causal effects of interventions across multiple units, incorporating counterfactual and synthetic control for policy evaluation in shared markets. This approach addresses potential spillover effects and leverages time series data for identification. We redefine causal effect estimation as predicting outcomes without intervention, first estimating counterfactual outcomes using high-dimensional time series data. This process utilizes cross-correlation in time series, employing an autoregressive recurrent neural network with parameter sharing. The second stage estimates and tests the average treatment effect on the target variable for statistical significance. Demonstrated through simulations and empirical studies, our method uniquely estimates effects using pre-treatment data in scenarios where traditional control unit assumptions fail. An empirical example estimates the impact of promotional deals on US grocery store sales, showcasing the method's applicability and contribution to existing literature.

Priscila Grecov: priscila.grecov@monash.edu
Klaus Ackermann: klaus.ackermann@monash.edu
Christoph Bergmeir: christoph.bergmeir@monash.edu

We thank seminar participants at Melbourne University, the University of Sydney, the TU Vienna, ETH Zürich and the University of Zürich, as well as conference participants at the International Panel Data Conference, Association for Applied Econometrics, and the Time Series and Forecasting Symposium. This work has benefited from comments from the following people in alphabetical order: Bin Peng, Giovanni Mellace, Didier Nibbering and Hasan Fallahgoul.

KEYWORDS: Counterfactual analysis, Global forecasting, Spillover effects.

1. INTRODUCTION

The problem of estimating the causal effects of a treatment or policy intervention over a time series of interest spans numerous areas, from public policy and governance to commercial applications. Traditional randomized controlled trials (RCT) or A/B testing are either not ethically possible or commercially feasible. In this context, there is a need for causal effect estimation using observational data.

We consider the problem of estimating the causal effect of a policy intervention in an observed set of time series with multiple treated units. These treated units are observed for T_0 periods before and T_* after some intervention occurs. In this group of time series, there is also additional information in the form of several untreated units, which serve as control instances and some covariates/external predictor time series. Based on this large dimensional panel of observed time series data, our idea is to use the information before the intervention period to forecast an artificial counterfactual outcome for the treated units by solving missing data on the potential outcome of a given treated unit by generating its series-specific counterfactual estimate. This synthetic counterfactual trend is then used to estimate the causal effect of the intervention.

Comparative case studies are ubiquitous in empirical research in social sciences, with different approaches for estimating the causal effects of the policy. Examples include the difference-in-difference (DiD) approaches (Athey and Imbens, 2006, Conley and Taber, 2011, Doudchenko and Imbens, 2016), synthetic control method (SCM) (Abadie et al., 2010), or a combination of DiD and SCM (Arkhangelsky et al., 2021), matrix completion (Athey et al., 2021), factor and interactive fixed effects (FE) models (Gobillon and Magnac, 2016), constrained lasso regression-based methods (Carvalho et al., 2018, Doudchenko and Imbens, 2016, Chernozhukov et al., 2021), Bayesian structural time series models (Brodersen et al., 2015, Klinenberg, 2021), and time series panel data models (Poulos, 2017, Chernozhukov et al., 2021). These methods are called counterfactual and synthetic control (CSC) methods in the causal inference literature. The basic idea behind all these methods is the Rubin Causal model (Rubin, 1974), called the *potential outcome* approach. In this potential outcome approach, causal statements related to a given intervention usually rely on estimating counterfactual outcomes based on time series from a similar group

of instances unaffected by this intervention. This paper presents an alternative method for performing the CSC-based counterfactual prediction – called **Deep Counterfactual Prediction Net (DeepCPNet)** – using deep neural networks (DNNs) in their global and recurrent versions. We examine scenarios where individual market actors respond to the actions of competing entities, such as switching to a different retailer in response to price cuts by a nearby competitor or choosing an alternative ride-sharing service due to cost advantages. However, our analysis concentrates on the aggregate outcomes of these individual decisions, in contrast to the individuals themselves (e.g. see (Bajari et al., 2021)).

In time series forecasting, the shift from *local* univariate models to *global* forecasting models (GFM) leverages data across multiple series for enhanced prediction accuracy (Bandara et al., 2020c, Januschowski et al., 2020, Salinas et al., 2020, Smyl, 2020, Montero-Manso and Hyndman, 2021). GFMs use a unified model for analyzing interrelated series, addressing sparse data and overfitting issues prevalent in *local* models. This global approach facilitates cross-learning from extensive datasets. Moreover, recognizing that autoregressive (AR) models with diverse lags can approximate simpler AR models, we adopt non-parametric neural network strategies to optimize mean predictions, integrating versatility in handling varying autoregressive processes.

In interconnected markets, isolating a control group unaffected by spillovers is challenging. Traditional counterfactual synthetic methods rely on control units or predictors unchanged by interventions both before and after the event. Recent studies, like those by Menchetti et al. (2023), Cerqua et al. (2022), suggest using only pre-event data from predictors, avoiding control series to bypass the problem of finding unaffected controls. However, this strategy overlooks valuable insights from related series generated under similar conditions, which are crucial for accurate comparisons and trend identification. Additionally, different sales time series may be influenced by similar underlying factors common across all series, making it essential to analyze data collected consistently across various contexts. For example, analyzing sales data collected consistently across different times, regions, or channels allows for meaningful trend comparisons, underscoring the significance of considering such homogeneous data and common underlying factors in analyses.

1.1. *The contributions of the article*

This study advances the counterfactual prediction literature in causal effect analysis, emphasizing high-dimensional panel data with multiple treated time series. It moves beyond classical synthetic control's need for distributional equivalency, leveraging similar underlying factors for identifying causal effects. Our approach is suited for complex time series with spillover effects and is confounding-resistant. We propose a CSC-based framework using GFM-DNN for forecasting counterfactuals in high-dimensional settings. Our focus is on the application of a GFM-DNN modelling to generate mean-unbiased proxies, $P_{i,t}^{(0)}$, for the counterfactual of i ($i > 0$) affected units (treated units) in the absence of the policy intervention, $Y_{i,t}^{(0)}$, in the post-treatment period ($t > T_0$):

$$Y_{i,t>T_0}^{(0)} = P_{i,t>T_0}^{(0)} + u_t, \quad E(u_t) = 0, \quad t = 1, ..., T_0 + T_*. \tag{1}$$

The treatment effect of a policy at time t is defined as $\delta_t = Y_{i,t>T_0}^{(1)} - Y_{i,t>T_0}^{(0)}$, where $Y_{i,t>T_0}^{(1)}$ represents the observed outcome for treated units post-policy intervention. Our focus is on estimating policy effects in the post-intervention period, so our set of interest is $\delta_t = \{\delta_t\}_{t=T_0+1}^{T_0+T_*}$. We frame the causal effect estimation challenge as a counterfactual forecasting task, aiming to achieve more accurate estimates $P_{i,t}^{(0)}$. Departing from traditional structural causal models—such as DiD, SCM, penalised and constrained Lasso regression estimators or Bayesian time series models, we advocate adopting GFM techniques leveraging deep neural networks. Only recently has the application of DNNs for counterfactual predictions gained traction in causal inference (Johansson et al., 2016, Hartford et al., 2017, Poulos, 2017, Lim, 2018, Shi et al., 2019, Farrell et al., 2021). These contributions underscore the advantages of DNNs' non-parametric nature, addressing conventional counterfactual prediction techniques' nonlinear and non-convex challenges. DNNs excel in capturing the temporal dynamics of time series and modelling nonlinear feature interactions and residuals without imposing stringent constraints on pre-intervention outcomes utilized during training, marking a distinct departure from extant models.

Contemporary counterfactual forecasting methods, including those using DNNs, often follow a *local* model approach, treating each series in panel data separately for future projections based on its history. This method's reliance on single time series over long periods can lead to issues, such as regime changes or structural breaks, complicating parameter es-

timation. To address these, modellers manually input prior knowledge like seasonal trends, relevant predictors, and key events, limiting adaptability.

In contrast, the *global* approach combines all time series to create a unified forecast function, assuming a common structure across series. Each series is normalized to remove fixed effects before training a single model on this dataset for predictions. After prediction, fixed effects are reintegrated. This method, akin to but distinct from pooling in econometrics, overcomes local model limitations by using a broader dataset. It utilizes the panel structure and factor interrelatedness, with higher-order nonlinear autoregressive processes approximating lower-order ones. As (Montero-Manso and Hyndman, 2021) notes, while local model complexity increases with more series, GFM's inherent complexity increases at a slower degree.

The GFM enhances CSC forecasting by needing only pre-treatment similarity between control and treated units, bypassing synthetic control method limitations by excluding post-treatment control data to avoid spillover effects. Its autoregressive nature boosts counterfactual prediction accuracy through lagged data from treated units. A global model facilitates placebo testing via control group forecasts, thus bolstering treatment estimate robustness. Additionally, we validate the consistency of global sieves estimators, reinforcing the method's credibility (Farrell et al., 2021, Fabozzi et al., 2019, Fallahgoul et al., 2024).

Our Monte Carlo simulation evaluates the asymptotic behavior of our methodology versus traditional techniques, focusing on nonlinear time series interactions and intervention heterogeneity. We assess model performance in scenarios with and without control unit spillovers. Results demonstrate DeepCPNet's superior performance and robustness in testing the null hypothesis of no intervention. We empirically measure the impact of promotional strategies on U.S. retail grocery sales, identifying key demographic responses. This approach benefits practitioners in interconnected markets, addressing challenges to the Stable Unit Treatment Value Assumption (SUTVA) due to market interconnectivity (Johari et al., 2022, Liu et al., 2021).

The plan of the article is orginized as follows. Section 2 presents the counterfactual methodology to estimate the treatment effect from benchmark models until the DeepCPNet proposal. This section also discusses the engine of the DeepCPNet framework, and the statistical tests performed to evaluate the significance of the results. The theoretical deliveries are presented in Section 3. In Section 4 we describe the simulated study and the empirical

exercise. In Section 5, we analyse the results obtained from the experiments, demonstrating the performance of DeepCPNet and discussing the main insights. Finally, Section 6 concludes the work and presents the future directions. Supplemental details, tables and figures are provided in the **Online Appendix** .

2. METHODOLOGY

2.1. *The counterfactual estimation problem*

We frame the counterfactual estimation problem as follows. Consider a collection of J time series representing units, which might be firms, products, states, countries, etc., indexed by $j = 1, ..., J$. Let the time period be denoted by $t = 1, ..., T_0, ..., T$, where $T_0 < T$. We posit that an intervention takes place at time T_0 for a subset i of these J units, such that ($n(i) > 0$ and $i \subset j$). Prior to T_0, the units in i are unaffected by the policy intervention, but post T_0, they are exposed to it for $T - T_0 = T_*$ periods. Typically, in the counterfactual framework, T_* is shorter than T_0. The units not subject to the policy intervention throughout the entire time horizon T are designated as control units, indexed by $c = 1, ..., C$, with $C < J$. The observable outcomes for these control units are represented by $Y_{c,t}$. The CSC methodology adopts the potential outcome framework as described by (Rubin, 1974, Neyman, 1923). Within this framework, the effect of a policy intervention on each treated unit at time t is characterized as the discrepancy between the potential outcomes influenced by the policy $Y_{i,t}^{(1)}$ and those unaffected $Y_{i,t}^{(0)}$, which represents the unobserved counterfactual requiring estimation, expressed as $\delta_{i,t} = Y_{i,t}^{(1)} - Y_{i,t}^{(0)}$ for $t > T_0$. Conversely, for control units, the observed outcomes $Y_{c,t}$ coincide with their outcomes in scenarios absence of the intervention, denoted as $Y_{c,t} = Y_{c,t}^{(0)}$ over the interval $1 \leq t \leq T$. The cumulative causal effect at the unit level during the policy's enactment is computed as $\sum_{t=T_0+1}^{T} \delta_{i,t}$. Furthermore, the *average treatment effect on the treated units (ATT)* is delineated, both at the unit level and aggregated over all treated units, as:

$$\Delta_i := \frac{1}{T - T_0} \sum_{t=T_0+1}^{T} \delta_{i,t} \Rightarrow \Delta := \frac{1}{J - C} \sum_{i} \Delta_i \,. \tag{2}$$

Building on these definitions, the CSC methodologies rest upon certain foundational assumptions, as elaborated by Chernozhukov et al. 2021, Cerqua et al. 2022, Menchetti et al. 2023.

ASSUMPTION 1—Counterfactual estimation using the potential outcome approach: *Suppose* $P_{i,t}^{(0)}$ *provides unbiased estimates for the counterfactual outcomes of treated units* $Y_{i,t}^{(0)}$ *in the absence of the treatment. Formally,* $E(P_{i,t}^{(0)}) = E(Y_{i,t}^{(0)})$*, and* $\delta_t = 0$ *for* $t \leq T_0$*. The potential outcomes for treated units under the intervention for* $(t > T_0)$ *are given by* $Y_{i,t}^{(1)} = Y_{i,t}^{(0)} + \delta_t$ *and can be expressed as:*

$$\left.\begin{array}{l} Y_{i,t}^{(0)} = P_{i,t}^{(0)} + u_t \\ Y_{i,t}^{(1)} = P_{i,t}^{(0)} + \delta_t + u_t \end{array}\right| \; with\ E(u_t) = 0,\ \ t = 1, ..., T$$

where u_t *is a stationary stochastic process, and the relation between the observed outcomes and potential outcomes is denoted by* $Y_{i,t} = Y_{i,t}^{(0)} + D_t(Y_{i,t}^{(1)} - Y_{i,t}^{(0)})$*, where* $D_t \in (0, 1)$ *is a binary indicator for the pre- or post-intervention period, such that* $D_t = 1$ *when* $t > T_0$ *and* $D_t = 0$ *otherwise.*

Assumption 1 presents the potential outcomes, positing the existence of unbiased proxies, $P_{i,t}^{(0)}$, without stipulating any specific dependencies on u_t. The assumption mandates u_t to be i.i.d. or, at a minimum, a stationary process exhibiting weak dependence. Crucially, the stochastic shock u_t is presumed invariant to the intervention, implying that the intervention's timing is orthogonal to factors influencing the distribution of u_t.

Consider a panel data set comprising treated series $Y_{i,t}$ and control series $Y_{c,t}$. For each unit j, which could be either treated or controlled, there exists an associated set of time series of external predictors denoted as $X_{j,t} = (x_{j,t}^1, ..., x_{j,t}^k) \in \mathbb{R}^k$, where $k > 0$ represents the covariates. Various modelling strategies exist for the estimation of counterfactual proxies $P_{i,t}^{(0)}$. Within our framework, $P_{i,t}^{(0)}$ is determined using solely the pre-treatment data $Y_{i,t \leq T_0}$, $Y_{c,t \leq T_0}$, and $X_{j,t \leq T_0}$ to estimate the parameters of the model. For simplification, and without loss of generality, we can assume that $Y_{j,t}, X_{j,t} \in \mathcal{Z} \subseteq \mathbb{R}^d$, where $d > 0$, and that $Y_{i,t}^{(0)} \in \mathcal{Y} \subseteq \mathbb{R}^q$, with $q > 0$.

$$Y_{i,t>T_0}^{(0)} = \mathcal{M}(Y_{j,t \leq T_0}, X_{j,t \leq T_0}, \theta) + u_t \tag{3}$$

where $\mathcal{M} : \mathcal{Z} \times \Theta \to \mathcal{Y}$ is a measurable mapping (the model) for each $\theta \in \Theta$, a finite dimensional parametric space.

However, in the current CSC literature, the functional form and inputs of $\mathcal{M}(\cdot,\cdot)$ can differ from (3) depending on the method chosen to predict the counterfactual $P_{i,t}^{(0)}$ for the treated units, as we will discuss in Section 2.2.

Another consideration under the CSC approach is that the potential outcomes for controls $Y_{c,t}$ are unaffected by the intervention at any time. Thus, the following fundamental assumption is also necessary.

ASSUMPTION 2—Null intervention effect over control units: *$Y_{c,t}$ is independent of D_t for all t, that is, $Y_{c,t} = Y_{c,t}^{(0)}$ for $1 \leq t \leq T$.*

Assumption 2 ensures that the control units remain unaffected by the intervention. This assumption is foundational to CSC methods and pivotal for discerning the intervention's effects. If $Y_{c,t}$ is independent of D_t, the treatment effect on treated units becomes identifiable given the deterministic and known timing of the intervention. Absent this independence, CSC-based counterfactual estimations would be compromised. Notably, this assumption's role diverges slightly from its traditional application in standard CSC methods. Given the independence between $Y_{c,t}$ and D_t, and our employment of the GFM (which shares learned parameters across all series j for prediction), the forecast of $Y_{c,t}$ acts as a validation metric, confirming the model's forecasting accuracy for times $t > T_0$. In the GFM framework, global parameters estimated during the pre-treatment phase are consistently utilized for post-intervention predictions for both control and treated time series. Consequently, we evaluate the model parameters' efficacy during the post-treatment period by testing for a null effect on controls.

In addition, we should ensure all treated units are subject to simultaneous and persistent intervention, as follows.

ASSUMPTION 3—Single Persistent Intervention: *Unit i receives a single persistent intervention when there exists a T_0 such that $D_t = 0$ for all $t \in \{1, \ldots, T_0\}$, but for all $t \in \{T_0 + 1, \ldots, T\}$ the unit is either always treated, $D_t = 1$, or always assigned to the control group, $D_t = 0$.*

Assumption 3 stipulates that the intervention is universally recognized, ensuring that all treated units are uniformly and continuously exposed to it for all $t > T_0$. This assumption also precludes any anticipatory effects related to the intervention.

In addition, the identification of the causal effect for time series setups necessitates the conditional stationarity assumption, consistent with findings from existing literature (Antonelli and Beck, 2020, Menchetti et al., 2023, Papadogeorgou et al., 2023, Cerqua et al., 2022). We articulate this assumption in the subsequent discussion.

ASSUMPTION 4—Conditional Stationarity for Identification: *Define* $\mathcal{I}_t = (\mathcal{I}_0, Y_{j,1}, ..., Y_{j,t}, X_{j,1}, ..., X_{j,t})$*, where* $\mathcal{I}_0$ *is the information setup to time* $t = 0$*. Let* $f_{Y^{(0)}_{j,t+1}|\mathcal{I}_t}$ *indicate the conditional density of the outcome from unit* j *at time* $t+1$ *given the past information such that* $f_{Y^{(0)}_{j,t+1}|\mathcal{I}_t}(y_{j,t+1}|\mathcal{I}_t)$ *is the conditional density in the absence of treatment. Then, for any integer* $t \geq 1$*,* $f_{Y^{(0)}_{j,t+1}|\mathcal{I}_t}\left(y_{j,t+1}\middle|\mathcal{I}_t\right) = f_{Y^{(0)}_{j,1}|\mathcal{I}_0}\left(y_{j,t+1}\middle|\mathcal{I}_t\right)$.

Given prior data, this assumption suggests the outcome's conditional distribution without treatment stays constant over time. Understanding $Y^{(0)}_{j,t}$'s distribution before intervention implies knowing it afterwards. Testing conditional stationarity involves checking pre-treatment forecast accuracy for unaffected control units (placebo test). Satisfactory performance indicates model validity, making post-intervention forecasts reliable for estimating counterfactual outcomes $Y^{(0)}_{i,t}$. This supports treatment effect identification from observational data as in Eq. (2).

A key feature of our method is adapting the Stable Unit Treatment Value Assumption (SUTVA) (Rubin, 1980) to allow for some interference among units. SUTVA traditionally requires no interference between units and no hidden treatment variations. Focusing on pre-intervention data, our approach relaxes the no-interference rule within a global forecasting framework. We require only a portion of control units to be unaffected by treatment for accurate point forecasts. Other controls may be influenced, yet their pre-treatment information remains central to our analysis.

The SUTVA requires outcomes to depend solely on the unit's treatment, with no external influences. Yet, its validity is challenged by potential spillovers among units, risking inaccurate causal conclusions. In addition to employing control units as covariates, the inclusion of external predictors associated with the outcome can enhance the precision of estimating $Y^{(0)}_{i,t}$. Yet, biases may emerge if treatments exert influence over these covariates or if there are spillovers impacting controls. By leveraging the Global Forecasting Mecha-

nism (GFM), we can circumvent these pitfalls by confining our analysis to the pre-treatment data of external predictors (as in Eq. (3)).

Moreover, our approach does not necessitate the external covariates to be insulated from treatment spillovers. However, in scenarios where post-treatment data from specific external covariates (e.g., holiday calendar dummies in our empirical study) are pivotal, these covariates must remain uninfluenced by the intervention. Consequently, we advocate for a more lenient rendition of SUTVA, which accommodates the estimation of causal effects even in the presence of potential spillover effects within panel data contexts.

ASSUMPTION 5—Weaker SUTVA: *Let* $Y_{1:C,1:T}$ *and* $X_{1:J,1:T}$ *be the entire panels of outcomes for the control units and of external covariates of all units, respectively. Let* $Y_{1:C^*,t>T_0} \subset Y_{1:C,1:T}$ *and* $X_{1:J^*,t>T_0} \subset X_{1:J,1:T}$ *be subsets of control units and external covariates. In panel settings where a single intervention happens at time* T_0 *and there are no treatment hidden forms, we assume the potential outcomes of the* c^**-th control unit* $Y_{c^*,t>T_0}$ *as the external covariate vector* $X_{j^*,t>T_0}$ *(if post-treatment data is used to predict the outcome of a unit) such subsets of external covariates and control units are independent of treatment interference, i.e.,* $X^{(1)}_{j^*,t} = X^{(0)}_{j^*,t}$ *and* $Y^{(1)}_{c^*,t} = Y^{(0)}_{c^*,t}$ *for all* $t \in \{T_0 + 1, \ldots, T\}$.

Therefore, Eq. (3) can be updated as follows:

$$Y^{(0)}_{i,t>T_0} = \mathcal{M}(Y^{(0)}_{j,t\leq T_0}, X^{(0)}_{j,t\leq T_0}, X^{(0)}_{j^*,t}, \theta) + u_t \tag{4}$$

2.2. *The choice of counterfactual estimation model*

The CSC method necessitates the selection of a model, denoted as $\mathcal{M}(\cdot,\cdot)$, designed to assimilate information from both treated and control units, as well as their covariates, based solely on pre-intervention data. This implies that the counterfactual is constructed using a model trained without any influence from the intervention. Building on the insights of Chernozhukov et al. (2021), the ensuing subsections provide an overview of prevailing CSC methodologies for $\mathcal{M}(\cdot,\cdot)$ aimed at estimating these counterfactual proxies. Subsequently, in Section 2.2.4, we delineate our innovative approach, leveraging feedforward neural networks (FFNNs) and recurrent neural networks (RNNs) for autoregressive time series forecasting within the ambit of the global forecasting paradigm.

2.2.1. *Difference-in-Difference Method*

The canonical DiD method introduced by Doudchenko and Imbens (2016) proposes a counterfactual proxy based on $P_{i,t}^{(0)} = \mu + \frac{1}{C}\sum_{c=1}^{C} Y_{c,t}^{(0)}$ and its estimation as $\hat{P}_{i,t}^{(0)} = \frac{1}{T}\sum_{s=1}^{T}\left(\frac{1}{J-C}\sum_{i=C+1}^{J} Y_{i,s}^{(0)} - \frac{1}{C}\sum_{c=1}^{C} Y_{c,s}^{(0)}\right) + \frac{1}{C}\sum_{c=1}^{C} Y_{c,t}^{(0)}$. We can observe that DiD estimation relies on a linear parallel trend hypothesis between the control and treated units. The assumption underlying the DiD method is that the level of the control unit group can be an adequate proxy for the level of the treated unit group in the absence of the intervention.

Variations of the DiD method, like those in Gobillon and Magnac (2016), extend it by using a linear panel model with exogenous regressors and interactive fixed effects. This requires correct model specification, estimation of common factors, and strict exogeneity of regressors, which is challenging in aggregate time series applications. Thus, we compare the canonical DiD method with our framework.

2.2.2. *Synthetic Control Method*

As detailed by (Abadie et al., 2010, Abadie, 2021), the canonical SCM is constructed under the presumption of a singular treated unit. In multiple treated units, separate SCM models must be fitted for each. This exposition focuses on the scenario with a single treated unit, denoted as $i = 1$. Assuming data for $C + 1$ units, indexed by $j = 1, ..., C + 1$, the first unit ($j = 1$) is designated as the treated unit, while the subsequent units $j = 2, ..., C + 1$ represent the control group. Absent any covariate considerations, the SCM stipulates the ensuing model:

$$P_{1,t}^{(0)} = \sum_{j=2}^{C+1} w_j Y_{j,t}^{(0)} \ , \ \textit{where } w \geq 0 \textit{ and } \sum_{j=2}^{C+1} w_j = 1 \, . \tag{5}$$

The method must assume the following condition to identify the weights w:

ASSUMPTION 6—Identifying assumption - SC: *The structural shocks u_t for the group of treated units do not have any correlation with the contemporaneous values of the outcomes, which means $E(u_t Y_{j,t}^{(0)}) = 0$ for $2 \leq j \leq C + 1$.*

Therefore, the counterfactual estimation is given by $\hat{P}_{1,t}^{(0)} = \sum_{j=2}^{C+1} \hat{w}_j Y_{j,t}^{(0)}$ where the SC estimator for w is as follows:

$$\hat{w} = \arg\min_{w} \sum_{t \leq T_0} \left(Y_{1,t}^{(0)} - \sum_{j=2}^{C+1} w_j Y_{j,t}^{(0)} \right)^2 \; s.t. \; w \geq 0 \; and \; \sum_{j=2}^{C+1} w_j = 1 \, . \tag{6}$$

Then, the estimated effect of the intervention of interest for the treated unit for the period $t > T_0$ is given by $\hat{\delta}_t = Y_{1,t}^{I} - \hat{P}_{1,t}^{(0)}$. The incorporation of the external predictors into the estimation problem (Eq. 6) is described in the **Online Appendix** (A.1.1).

While SCM addresses DiD's limitations, it assumes covariates' convex combination models the counterfactual, potentially biasing estimates ((Ferman and Pinto, 2016, Carvalho et al., 2018)). It disregards nonconvex and nonlinear relationships among controls, assuming a perfect pre-treatment fit. SCM's reliance on temporal averages removes time series dynamics and lacks a method for selecting variables critical for optimal weight determination.

In contrast to the SCM, in the **Online Appendix**, we also discuss three additional and more flexible CSC approaches which relax the strong non-negativity on the weights: the *constrained LASSO method* (A.1.2), the *Penalized Regression Method* (A.1.3) and the *Bayesian Structural Time Series Models* (A.1.4), with the first and the latter being considered as one of the benchmark models in our experiments, the ArCo model developed by Carvalho et al. (2018) and the CausalImpact model introduced by Brodersen et al. (2015), respectively.

2.2.3. *Summary of CSC methods discussed previously*

SCM and all previously presented structural causal models construct the counterfactual based on post-treatment observed variables from a pool of covariates, encompassing control units and/or external predictors. Therefore, a crucial assumption for these existing CSC approaches is the constancy of the relationship between the covariates $\mathbf{x}_1, ..., \mathbf{x}_t$ and each treated time series $y_1, ..., y_t$ modelled during the pre-intervention period, extending into the post-intervention period as follows:

$$y_t = \alpha_0 + \beta_t^{'} \mathbf{x}_t + \varepsilon_t, \; \forall t = 1, ..., T_0 \, , \tag{7a}$$

$$\hat{y}_t = \alpha_1 + \hat{\beta}_t^{'} \mathbf{x}_t + \zeta_t, \; \forall t = T_0 + 1, ..., T \, , \tag{7b}$$

$$\hat{\delta}_t = y_t - \hat{y}_t, \; \forall t = T_0 + 1, ..., T. \tag{7c}$$

Provided that the covariates $\mathbf{x}_t$ remain unaffected by the intervention, as stipulated in Assumption 2, all the CSC models discussed herein presuppose the maintenance of the relationship as mentioned earlier post-intervention. Consequently, the model calibrated in Eq. (7a) is utilized to predict the counterfactual time series for treated units after the intervention, as delineated in Eq. (7b). The estimated treatment effect is subsequently derived as the discrepancy between the observed time series of the treated units and their forecasted counterfactual counterparts, as expressed in Eq. (7c).

Existing CSC methods assume a consistent relationship between treated units and predictors before and after treatment, which is essential for valid counterfactual predictions. Our framework, however, reduces reliance on this assumption by focusing solely on pre-treatment data, needing consistency only until intervention.

2.2.4. *Autoregressive Global Time Series Models*

2.2.4.1. *Time series forecasting (local and global approaches)* Autoregressive time series forecasting involves predicting future values of a time series based on its historical data. Consider a simple example with a target series $Y = \{y_1, y_2, ..., y_t, ..., y_T\}$. The forecasting problem in an autoregressive context can be expressed as $\{y_{T+1}, ..., y_{T+H}\} = F(y_1, y_2, ..., y_T) + \epsilon$, where F represents the function modelled to predict future series values from time step $T+1$ to $T+H$, H is the forecasting horizon, and ϵ denotes the error associated with the approximation function F.

Consider a set $\mathbb{S}$ comprising J univariate time series, denoted as $\mathbb{S} = \{Z_j \in \mathbb{R}^{T_0},\ j = 1, ..., J\}$, where Z_j represents the observed time series with length T_0 corresponding to the temporal dimension. We assume our objective is to predict a finite number of future time steps, T_* (the forecast horizon). The forecasting functions can be formally defined as mappings from the space of past observations to future values, specifically $f : \mathbb{R}^{T_0} \to \mathbb{R}^{T_*}$.

Now, consider a local learning algorithm A_L that takes a single time series Z_j as input and yields a corresponding forecasting function f_j, formally defined as $A_L : \mathbb{R}^{T_0} \to (\mathbb{R}^{T_0} \to \mathbb{R}^{T_*})$, where $T_* = T - T_0$. In this framework, the algorithm A_L is applied individually to each time series Z_j in the set $\mathbb{S}$, generating a distinct forecasting function f_j tailored to each series. Conversely, a global learning algorithm A_G operates on the entire set of time series $\mathbb{S}$ and produces a single, universal forecasting function g, which is represented as $A_G : \mathbb{S} \to (\mathbb{R}^{T_0} \to \mathbb{R}^{T_*})$.

This research proposes a framework that leverages a global autoregressive modelling approach for time series forecasting to estimate the unobserved counterfactual outcomes of treated units. In this method, data from all types of time series – control, treated, and external predictors – are pooled together to fit a single, comprehensive univariate forecasting function.

Revisiting the structure of the counterfactual estimation problem outlined in Assumption 1 and integrating it with our global autoregressive time series forecasting approach, we reformulate the counterfactual estimation problem as follows:

$$\left.\begin{aligned} Y_{i,t}^{(0)} &= g\big(Z_{j,t-1}^{(0)}\big) + u_t \\ Y_{i,t}^{(1)} &= g\big(Z_{j,t-1}^{(0)}\big) + \theta_t + u_t \end{aligned}\right| \; with\; E(u_t) = 0,\; \{u_t\}\; iid,\;\; t = 1, ..., T_0, ..., T \tag{8}$$

where the unbiased estimation for the counterfactual $Y_{i,t}^{(0)}$ is provided by $P_{i,t}^{(0)} = g\big(Z_{t-1}^{(0)}\big)$. Here, g represents an autoregressive model function derived from the global algorithm A_G. For a basic autoregressive model such as AR(p), $g(\cdot)$ takes the form $\sum_{p=0}^{P} g_p L^p(\cdot)$, where L denotes the lag operator. In scenarios involving neural network models applied to time series data, $g(\cdot)$ becomes a nonlinear function of lag operators, exemplified as $g(\cdot) = m(\cdot, L^1(\cdot), ..., L^p(\cdot))$. The series $Z_j^{(0)} \in \mathbb{S}$ comprises the treated time series $Y_i^{(0)}$, the control time series $Y_c^{(0)}$, and the external predictors $X_j^{(0)}$, with $i \cup c = j$. Each of these time series, of length T_0 (the intervention time), is transformed through lag-embedding into a matrix of a given AR order (p), resulting in the matrices $\mathbf{Y}_i$, $\mathbf{Y}_c$, and $\mathbf{X}_j$. These matrices are then concatenated to form a large matrix $\mathbf{Z}^{(0)} = \big[\mathbf{Y}_i\, \mathbf{Y}_c\, \mathbf{X}_j\big]'$, encompassing $2 \times J$ time series, serving as the pooled input data. The model minimizes a loss function across this comprehensive global matrix, analogous to a standard regression problem. Consequently, given $\big\{\mathbf{Z}^{(0)}\big\}_{t \le T_0}$, our model endeavours to learn a nonlinear mapping $g(\cdot)$ that can predict the counterfactual values for each treated time series from $T_0 + 1$ to $T_0 + H$, expressed as $\{P_{i,T_0+1}^{(0)}, ..., P_{i,T_0+H}^{(0)}\} = g(Z_1^{(0)}, Z_2^{(0)}, ..., Z_{T_0}^{(0)})$.

As illustrated, our autoregressive estimation approach does not depend on post-treatment data for predicting the counterfactual, in contrast to conventional causal models, such as represented in Eq. (7b). Consequently, our framework does not necessitate the preservation of the pre-treatment relationship between treated units and controls in the post-treatment period, nor does it require strict adherence to the Stable Unit Treatment Value Assumption (SUTVA); instead, we can apply a more lenient version as delineated in Assumption 5. This

affords our model greater flexibility, allowing for potential post-intervention disturbances in the control series without compromising the accuracy of the counterfactual predictions, even under the assumption of a null intervention effect on the control units (Assumption 2).

Adopting an autoregressive Global Forecasting Model (GFM) in our framework offers a distinct advantage, permitting the incorporation of treated time series data alongside control series and covariates. This comprehensive inclusion of data enhances the model's counterfactual forecasting capabilities. The global model $g(\cdot)$ can be any autoregressive GFM in this context. Specifically, we advocate for the use of a deep neural network (DNN), such as a Feedforward Neural Network (FFNN) or a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) architecture, as our global model. In subsequent sections, we will explore various candidate models for $g(\cdot)$ and elucidate the advantages of selecting the RNN-LSTM algorithm as a pivotal component of our framework.

2.2.4.2. *Nonlinear and non-parametric models* The structural, regression causal models, and autoregressive models outlined in Sections 2.2.1-2.2.3 are predominantly parametric and linear in nature. However, real-world phenomena are often nonlinear, necessitating the exploration of nonlinear and non-parametric models for more accurate causal effect estimation. Linear regression models, while widely used, assume an underlying linear relationship between the response variable and its predictors, and they require the error distribution of the target variable to be normally distributed.

In the **Online Appendix** (A.1.5 and A.1.6), we will elucidate how Generalised Linear Models (GLM), Generalised Additive Models (GAM) and Threshold Autoregressive (TAR) models achieve nonlinearity in modelling, their nonlinear applications, and limitations to understand better the benefits of adopting a neural learning-based approach for our global model $g(\cdot)$.

Our approach uses a higher autoregressive order with more lagged variables to capture nonlinearities, requiring a complex model class. NNs are preferred for global modelling over GAMs due to their superior handling of complex nonlinear interactions.

2.2.4.3. *DNN-FFNN models for time series forecasting* Deep neural networks (DNNs) effectiveness relies on balancing model complexity and training data volume. Overfitting occurs when a model for the data is too complex, whereas underfitting happens when there is insufficient architecture for the data volume. Our framework employs deep architectures

of feedforward neural networks (FFNNs) and Recurrent Neural Networks (RNNs) as outlined in our sequence.

The FFNN model constitutes the fundamental form of NN, characterized by a unidirectional flow of data, whereby information is processed in only one direction, without recurrence, in contrast to a RNN. Let $\left\{\left(Y_t, \mathbf{X}_t^\top\right)\right\}_{t=1}^{\infty}$ be a covariance-stationary stochastic process taking values on $\mathbb{R}^{d+1}$ and $\mathbf{V}_t \in \mathbb{R}^n$ a vector of lags and predictors such as $\mathbf{V}_t = \left(Y_{t-1}, ..., Y_{t-p}, ..., \mathbf{X}_t^\top, ..., \mathbf{X}_{t-r}^\top\right)^\top$ where $n = p + d(r+1)$. An autoregressive and shallow (one layer) FFNN of order p, for $t = 1, ..., T$ and $h = 1, ..., H$, is of the form $Y_{t+h} = f_\Theta(\mathbf{V}_t) + \epsilon_t$, where $\mathbb{E}(\epsilon_t) = 0$, Θ is the set of parameters and $f_\Theta : \mathbb{R}^n \to \mathbb{R}$ is the FFNN architecture where the inputs consist of lagged values of the target Y_t and the covariates $\mathbf{X}_t$, i.e.

$$f_\Theta(\mathbf{V}_t) := f(\mathbf{V}_t; \Theta) = \beta_0 + \sum_{k=1}^{K} \beta_k \phi(\mathbf{w}_k^\top \mathbf{V}_t + w_{0,k}) = \beta_0 + \sum_{k=1}^{K} \beta_k \phi(\tilde{\mathbf{w}}_k^\top \tilde{\mathbf{V}}_t), \tag{9}$$

where $\phi(\cdot)$ is the activation function, $w_k \in \mathbb{R}^n$ for $k = 1, ..., K$ are the weight vector between the input data and the hidden nodes; β_k are the coefficients between the latent variables from the hidden layer and the output; and the $\beta_0, w_{0,k} \in \mathbb{R}$ are bias coefficients associated to the output and hidden nodes, respectively. This set of parameters we denote $\Theta = (\beta_0, ..., \beta_K, \mathbf{w}_1^\top, ..., \mathbf{w}_K^\top, w_{0,1}, ..., w_{0,K})$, which is the vector to be estimated. An illustration of this architecture can be seen in Fig. A.7 in the **Online Appendix** (A.7). If we assume $\tilde{\mathbf{V}}_t = (1, \mathbf{V}_t^\top)^\top$ and $\tilde{\mathbf{w}}_k = (w_{0,k}, \mathbf{w}_k^\top)^\top$, we have the reduced form of Eq. 9. Hence, model (9) can be rewritten in a matrix notation. Let $\mathbf{W} = (\tilde{\mathbf{w}}_1, ..., \tilde{\mathbf{w}}_K)$,

$$\mathbf{V} = \begin{pmatrix} 1 & V_{11} & \cdots & V_{1p} \\ 1 & V_{21} & \cdots & V_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & V_{T1} & \cdots & V_{Tp} \end{pmatrix}, \text{ and } \boldsymbol{\mathcal{O}}(\mathbf{VW}) = \begin{pmatrix} 1 & \phi(\tilde{\mathbf{w}}_1^\top \tilde{\mathbf{x}}_1) & \cdots & \phi(\tilde{\mathbf{w}}_K^\top \tilde{\mathbf{x}}_1) \\ 1 & \phi(\tilde{\mathbf{w}}_1^\top \tilde{\mathbf{x}}_2) & \cdots & \phi(\tilde{\mathbf{w}}_K^\top \tilde{\mathbf{x}}_2) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \phi(\tilde{\mathbf{w}}_1^\top \tilde{\mathbf{x}}_T) & \cdots & \phi(\tilde{\mathbf{w}}_K^\top \tilde{\mathbf{x}}_T) \end{pmatrix}.$$

Therefore, by defining $\boldsymbol{\beta} = (\beta_0, \beta_1, \cdots, \beta_K)^\top$, the output of a FFNN is given by:

$$\boldsymbol{f}(\boldsymbol{V}; \Theta) = \left[f(\boldsymbol{V}_1; \Theta), \cdots, f(\boldsymbol{V}_T; \Theta)\right] = \boldsymbol{\mathcal{O}}(\mathbf{VW})\boldsymbol{\beta}. \tag{10}$$

We can now extend the notation for the case of deep (multi-layer) FFNNs. Set K_l as the number of hidden nodes in layer $l \in \{1, \cdots, L\}$. For each hidden layer l, define $\boldsymbol{W}_l = (\tilde{\mathbf{w}}_{1l}, \cdots, \tilde{\mathbf{w}}_{Kl})$. Then, the output $\boldsymbol{\mathcal{O}}_l$ of layer l can be written as $\boldsymbol{\mathcal{O}}_l(\boldsymbol{\mathcal{O}}_{l-1}(\cdot)\boldsymbol{W}_l) =$

$\begin{pmatrix} 1 & \phi(\tilde{\mathbf{w}}_{1l}^{\top}\boldsymbol{\mathcal{O}}_{1l-1}(\cdot)) & \cdots & \phi(\tilde{\mathbf{w}}_{Kl}^{\top}\boldsymbol{\mathcal{O}}_{1l-1}(\cdot)) \\ 1 & \phi(\tilde{\mathbf{w}}_{1l}^{\top}\boldsymbol{\mathcal{O}}_{2l-1}(\cdot)) & \cdots & \phi(\tilde{\mathbf{w}}_{Kl}^{\top}\boldsymbol{\mathcal{O}}_{2l-1}(\cdot)) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \phi(\tilde{\mathbf{w}}_{1l}^{\top}\boldsymbol{\mathcal{O}}_{nl-1}(\cdot)) & \cdots & \phi(\tilde{\mathbf{w}}_{Kl}^{\top}\boldsymbol{\mathcal{O}}_{nl-1}(\cdot)) \end{pmatrix}$, where $\boldsymbol{\mathcal{O}}_0 := \boldsymbol{X}$. Therewith, the output of the DNN-FFNN is the composition of

$$\boldsymbol{f}(\boldsymbol{V}) = \boldsymbol{\mathcal{O}}_L(\cdots \boldsymbol{\mathcal{O}}_3(\boldsymbol{\mathcal{O}}_2(\boldsymbol{\mathcal{O}}_1(\mathbf{V}\mathbf{W}_1)\mathbf{W}_2)\mathbf{W}_3)\cdots)\mathbf{W}_L\boldsymbol{\beta}. \tag{11}$$

2.2.4.4. *DNN-RNN-LSTM models for time series forecasting* RNNs, a subclass of DNNs for sequence modeling, have *memory states* enabling them to retain and propagate sequential information via recurrent connections. These features allow RNNs to maintain temporal dependencies by using past inputs and shared model parameters. Despite their advantages, RNNs, as introduced by Elman (1990), face issues with vanishing and exploding gradients, limiting their ability to handle long-term dependencies in long sequences and risking instability in weight matrices. To overcome these limitations, Hochreiter and Schmidhuber (1997) developed the Long Short-Term Memory (LSTM) network, enhancing RNNs' capacity for sequence learning. A detailed exposition of RNN and LSTM models is available in the **Online Appendix** (A.2).

2.2.4.5. *Global-FFNN and Global-LSTM models for time series forecasting* Following Bandara et al. (2020c) and Hewamalage et al. (2021b), our forecasting model utilizes FFNN or LSTM networks in a stacked configuration, where each unit is repeated across time steps with consistent weights and biases. This setup layers multiple FFNN or LSTM units, with each layer's output feeding into the next. An affine neural layer on top ensures output dimensions match our forecasting horizon. Parameter optimization employs the Adam optimizer (Kingma and Ba, 2014), adjusting learning rates adaptively during training.

The input to our stacked Global-FFNN and Global-LSTM models is the comprehensive and concatenated global matrix $\mathbf{Z}^{(0)} = \begin{bmatrix} \mathbf{Y}_i & \mathbf{Y}_c & \mathbf{X}_j \end{bmatrix}'$, which encompasses the time series data for treated units, control units, and external covariates, respectively. The model is trained using the data from the pre-treatment period $\left\{\mathbf{Z}^{(0)}\right\}_{t \leq T_0}$. Specifically, given a set of J univariate time series $\{y_{j,1:T_0}\}_{j=1}^{J}$, which includes all I treated and C control units ($J = I + C$), our network aims to model the future trajectories $\{y_{j,T_0+1:T}\}_{j=1}^{J}$ of each time

series following the intervention, based on their pre-intervention data. The pre-intervention data from both treated and control units, along with the associated external covariate vectors $\{\mathbf{x}_{j,1:T_0}\}_{j=1}^{J}$, serve as the inputs to our global network. This relationship is formally expressed as $\{y_{j,T_0+1:T}\}_{j=1}^{J} = g(y_{:j,1:T_0}, \mathbf{x}_{:j,1:T_0}, \Theta)$, where g denotes our *global* network model, Θ denotes the set of learnable parameters of the network model, $y_{j,t} \in \mathbb{R}$ is the value of the j-th time series at time t, and $\mathbf{x}_{j,t} \in \mathbb{R}^k$ is an external covariate vector.

To provide a comprehensive overview, our modelling framework operates as follows: each time series $y_{j,t}$ from the pre-intervention periods of treated and control units, $\mathbf{Y}_i \cup \mathbf{Y}_c$, along with optional external covariates $\mathbf{x}_{j,t}$ from $\mathbf{X}_j$, are fed into our Global-FFNN or Global-LSTM models (which architecture of the latter adheres to that described in the **Online Appendix** (A.2)), and both functions in an autoregressive manner. In particular, the LSTM also functions in a recurrent manner, taking into account the values from the previous time step as well as the prior state of the network. Consequently, the network output $\mathbf{h}_{j,t}$—which represents the post-intervention predictions for each series—of our autoregressive NN *global* model $g(\cdot)$, as characterized by its hidden states (h_t), can be mathematically formulated as:

$$\mathbf{h}_{j,t}^{FFNN} = g(y_{:j,t-1}, \mathbf{x}_{:j,1:T_0}, \Theta), \tag{12a}$$

$$\mathbf{h}_{j,t}^{LSTM} = g(\mathbf{h}_{j,t-1}, y_{:j,t-1}, \mathbf{x}_{:j,1:T_0}, \Theta). \tag{12b}$$

In Eqs. (12), the function $g(\cdot)$ represents a *global* and multi-layer FFNN or LSTM model, parameterized by Θ, and incorporating a stacked architecture. This design choice aligns with the recommendations provided by Bandara et al. (2020c), Hewamalage et al. (2021b) and is tailored to support our *global* modelling approach. Following the principles of the Global Forecasting Model (GFM) approach, the learnable parameters encapsulated in Θ are shared across all J time series, ensuring a joint learning process. This shared parameterization enriches the model with a wealth of information from all-time series, thereby enhancing the accuracy and reliability of our counterfactual forecasting.

2.3. *The DeepCPNet forecast framework*

In the initial phase of our DeepCPNet framework, we employ the *global* FFNN (DeepCPNet-FFNN) or LSTM (DeepCPNet-LSTM) model $g(\cdot)$ to forecast post-intervention outcomes for all units $j = 1, ..., J$, which encompasses both the counterfactual outcomes

for the treated units $i = 1, ..., I$ and the factual outcomes for the control units $c = 1, ..., C$. This process utilizes a comprehensive set of time series data from treated, untreated, and regressor variables, as delineated in Eqs. (12). In this context, referring back to Eq. (3), the mapping function $\mathcal{M}(\cdot)$ is instantiated by our proposed model $g(\cdot)$, resulting in the following expression:

$$\{Y^{(0)}_{i,t>T_0}, Y^{(0)}_{c,t>T_0}\} = g(Y_{j,t\leq T_0}, \mathbf{X}_{j,t\leq T_0}, \Theta) + u_t, \text{ where } j = 1, \ldots, I, I+1, \ldots, C.$$

Subsequently, in the second stage, we estimate the average treatment effects on the treated and control units, as defined in Equation (2). This estimation is formalized as follows:

$$ATT = \frac{1}{I}\sum_{i=1}^{I}\frac{1}{T-T_0}\sum_{t=T_0+1}^{T} Y^{(1)}_{i,t} - \hat{Y}^{(0)}_{i,t}, ATE_{ctr} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{T-T_0}\sum_{t=T_0+1}^{T} Y^{(1)}_{c,t} - \hat{Y}^{(0)}_{c,t}, \quad (13)$$

where, $Y^{(1)}_{i,t}$ and $Y^{(1)}_{c,t}$ represent the observed post-intervention outcomes for the treated and control units, respectively. In addition to estimating the treatment effects, the second stage of our framework also includes statistical tests to assess the significance of the treatment effect on the treated units and placebo tests to evaluate the null effect of the treatment on the control units, serving as a measure of the accuracy of our counterfactual outcome estimations. These procedures are elaborated upon in the subsequent subsections.

2.3.1. *Stage 1: Forecasting engine of DeepCPNet*

The forecasting engine of DeepCPNet encompasses three distinct layers, arranged sequentially as follows: 1) the preprocessing layer, 2) the DeepCPNet training layer, and 3) the post-processing layer.

The preprocessing layer starts with normalizing time series via mean-scale transformation to reduce variation and highlight the intervention effect. It addresses series-specific fixed effects, aiding in estimating the policy intervention's within-effect. Logarithmic transformation is then applied to stabilize variance and ensure additive seasonality and trends, a common practice in forecasting. Seasonal components are extracted using the Seasonal and Trend Decomposition using Loess (STL) method (Cleveland et al., 1990), later integrated as exogenous variables in the DeepCPNet model training.

In the training phase, time series data is divided into fixed-size windows using the moving window transformation strategy, generating multiple training instances to enhance model generalization. For each window, the mean absolute error (MAE) is calculated as the loss function and minimized using the Adam optimizer (Kingma and Ba, 2014). The dataset is split into training, validation, and test/forecast subsets, with the validation set used for hyper-parameter tuning via the Sequential Model-Based Configuration (SMAC) algorithm (Hutter et al., 2011), employing a Bayesian optimization approach (Hewamalage et al., 2021b). The Seasonal Exogenous (SE) training approach, proposed by Bandara et al. (Bandara et al., 2020a), incorporates seasonal components as exogenous variables in training, providing an additional mechanism for DeepCPNet to account for seasonality.

In our framework's final stage, the post-processing layer reverses pre-processing transformations, restoring model predictions to the original data scale, as illustrated in Fig. A.4 for DeepCPNet. TensorFlow™ version 2.0, an open-source deep-learning toolkit, is used for implementation (Abadi et al., 2015, Hewamalage et al., 2021b).

2.3.2. *Stage 2: Inference of treatment effect and placebo tests*

To infer the causal effects of the treatment and conduct placebo tests, we adhere to the hypothesis testing framework outlined by Chernozhukov et al. (2021). In their seminal work, Chernozhukov et al. (2021) introduce innovative inference procedures that are broadly applicable across various synthetic control methods (SCMs) used in policy evaluation. Specifically, these techniques are designed to predict counterfactual mean outcomes in the absence of a policy intervention, which requires fulfilling Assumption 1 where $P_{i,t}^{(0)}$ is estimated under the null hypothesis. This is crucial for achieving good small sample performance. Without this, if $P_{i,t}^{(0)}$ is estimated using pretreatment data without imposing the null, permuting the block of residuals does not yield procedures with exact finite sample validity, even with iid data. Within this framework, we focus on testing the null hypothesis of no treatment effect, formulated as follows:

$$\begin{cases} H_0 : \hat{\Delta} = 0, H_A : \hat{\Delta} \neq 0, \end{cases} \tag{14}$$

where $\hat{\Delta}$ is the estimated average treatment effect, as defined in Eq. (13).

For the treated units, as quantified by $\hat{\Delta}_{\text{treated}}$ or the Average Treatment Effect on the Treated (ATT), our objective is to reject the null hypothesis H_0 in Eq. (14), which would

provide substantial evidence of a positive impact from the policy intervention. Conversely, for the control units, quantified by $\hat{\Delta}_{\text{controls}}$, we aim to accept H_0, thereby validating Assumption 2 that posits a null intervention effect on the control units—this constitutes our placebo test. In essence, the placebo test involves comparing the distribution of prediction errors for the treated units (reflecting the treatment effect) with that of the control units (reflecting post-intervention forecast trajectories). If the policy has had an impact, the prediction errors for the treated units should be substantially larger, indicating a treatment effect different from zero. Conversely, the prediction errors for the control units should align with the null effect hypothesis. To formalize this, we define the test statistic and p-value as follows:

$$S(\hat{u}) = \left(T_*^{-1/2} \sum_{t=T_0+1}^{T} |\hat{u}_t|\right), \tag{15}$$

$$\hat{p} = 1 - \hat{F}\big(S(\hat{u})\big), with \hat{F}(x) = \frac{1}{\Pi} \sum_{\pi \in \Pi} \mathbf{1}\big\{S\big(\hat{u}_\pi\big) < x\big\}, \tag{16}$$

where $\hat{u}_t$ represents the prediction error, defined as the discrepancy between observed and predicted values, encapsulating the treatment effect δ_t (for a comprehensive discussion, see Chernozhukov et al. (2021)). However, it is crucial to note that while Chernozhukov et al. (2021) focused on scenarios with a single treatment unit, leading to a univariate time series for the test ($\hat{u} \in \mathbb{R}^t$), our study extends this framework to accommodate multiple treated units. Given our interest in assessing the aggregate impact of a policy across a group of units (whether treated or control) over the entire post-intervention period, as encapsulated in Eq. (13), we reformulate the test statistic to account for multiple series. Consequently, in our context, $\hat{u} \in \mathbb{R}^{j \times t}$, leading to a modification of the test statistic in Eq. (15) to $S(\hat{u}) = \left(J^{-1/2} \sum_{j=1}^{J} T_*^{-1/2} \sum_{t=T_0+1}^{T} |\hat{u}_{j,t}|\right)$.

3. THEORETICAL RESULTS

This section is dedicated to establishing the consistency of *global* Feedforward Neural Network (FFNN) models as estimators, building upon and extending the results from prior research. We aim to generalize the proof of consistency for *local* FFNN models provided by Fallahgoul et al. (2024) to *global* FFNNs. Additionally, we revisit and formalize the

arguments presented by Montero-Manso and Hyndman (2021), demonstrating the superior accuracy of *global* models compared to *local* models.

To structure our discussion, we start in Subsection 3.1 by presenting a refined proof of the results from Montero-Manso and Hyndman (2021), explicitly addressing the enhanced accuracy of *global* models over their *local* counterparts (see Propositions 1 and 2). Following that, in Subsection 3.2, we summarize the key findings of Fallahgoul et al. (2024) concerning the consistency of *local* FFNN estimators (refer to Theorem 1).

Building upon these foundations, we proceed to Subsection 3.3 where we amalgamate the insights from Fallahgoul et al. (2024) (Theorem 1) with our enhanced understanding of Proposition 2 from Montero-Manso and Hyndman (2021). This synthesis culminates in establishing the consistency of *global* FFNN models, encapsulated in Corollary 1.1, representing this paper's central theoretical contribution. According to the results presented in Carvalho et al. (2018), once the counterfactual estimator is consistent, the ATE estimator that utilizes this consistent counterfactual estimator also becomes consistent.

3.1. *The relation between global and local approaches for forecasting a set of time series*

This subsection delves into the relationship between global and local forecasting methods, shedding light on the formal principles that underscore the enhanced performance of global approaches, as established by Montero-Manso and Hyndman (2021) and further explored in our work. Initially, Montero-Manso and Hyndman demonstrate that the individual forecasts generated by a local method's distinct functions can be equivalently produced by a single, comprehensive function utilized by a global algorithm, as formalized in Proposition 1. Subsequently, they establish that as the size of the sample set increases, the complexity of local algorithms escalates accordingly, eventually exceeding the relatively constant complexity of global algorithms. The latter's complexity remains stable, provided that the number of parameters within its singular forecasting function is maintained, as stated in Proposition 2. Consequently, given an equivalent sample set size, global models have the capacity to incorporate enhanced complexity—whether through the addition of features, extended memory, or other means—thereby yielding improved accuracy and superior generalization compared to local models.

3.1.1. *Equivalence of global and local methods*

Utilizing the definitions for global (A_G) and local (A_L) algorithms, as well as the set of univariate time series $\mathbb{S}$ provided in Section 2.2.4, we present the following proposition.

PROPOSITION 1—Equivalence of Local and Global Algorithms for Time Series Forecasting: *Let $\mathbb{A}_L$ = set of all local learning algorithms, and $\mathbb{A}_G$ = set of all global learning algorithms . Then,*

1. *For any $A_G \in \mathbb{A}_G$, there exists an $A_L \in \mathbb{A}_L$ such that $\big(A_G(\mathbb{S})\big)(Z_j) = A_L\big(Z_j\big) \forall Z_j \in \mathbb{S}$*
2. *For any $A_L \in \mathbb{A}_L$, there exists an $A_G \in \mathbb{A}_G$ such that $A_L\big(Z_j\big) = \big(A_G(\mathbb{S})\big)(Z_j) \forall Z_j \in \mathbb{S}$*

PROOF: Montero-Manso and Hyndman (2021) prove Proposition 1 as follows:

1. Given the function $g = A_G(\mathbb{S})$, an algorithm A_L can be defined such that each $f_j = g$, produces identical forecasts regardless of the input series Z_j.
2. In this case, the output of A_L, denoted as $\big\{f_j, j = 1, ..., J\big\}$, is fixed. The authors demonstrate the existence of a function g capable of approximating these outputs. Two scenarios are considered:

   (a) when $Z_j \neq Z_i$ for all $j \neq i$: Every $Z_j \in \mathbb{S}$ is unique. Given the finite nature of $\mathbb{S}$, the set $f_j(Z_j)$ is also finite, with cardinality at most J. Consequently, there exists a function g mapping from a finite set to another finite set of equal or smaller size (codomain of g) $\leq$ (domain of g). According to the pigeonhole principle, there exists a function that maps each element in the codomain to at least one element in the domain.

   (b) when $Z_j = Z_i$ for some $j \neq i$ but $f_j(Z_j) \neq f_i(Z_i)$: No global function g exists since it would require multiple outputs for the same input, contradicting the definition of a function. Montero-Manso and Hyndman (2021) highlight that global methods are more restrictive than local methods in general regression problems, where it is possible to have $Z_j = Z_i$ but $y_j \neq y_i$. In such scenarios, local methods could find functions f_j and f_i such that $f_j(Z_j) = y_j$ and $f_i(Z_i) = y_i$, but no single function g could produce this mapping. However, in time series forecasting, where Z_j is the unique input to the algorithm, this situation is not possible. For a

local algorithm A_L, if $Z_j = Z_i$, then $A_L(Z_j) = f_j = A_L(Z_i) = f_i$, ensuring that both scenarios produce a valid function g.

Q.E.D.

In the online appendix in section A.3, we illustrate the results for AR processes.

3.1.2. *Generalisation relationship between global and local methods*

In the context of forecasting, our primary goal is to generate predictions that are accurate for unseen (out-of-sample) data, aiming to find an algorithm that exhibits strong generalisation properties. In this context, generalisation refers to the algorithm's ability to minimize the *generalisation gap*, which is the discrepancy between out-of-sample and in-sample errors. Proposition 1 established that both local and global algorithms have the potential to yield identical forecasts, potentially achieving equivalent levels of accuracy. Nonetheless, this assertion does not inherently provide a practical framework for comparing the generalisation performance of two given algorithms.

Denoting the out-of-sample error as E_{out} and the in-sample error as E_{in}, Montero-Manso and Hyndman (2021) elucidated the relationship between these errors. Specifically, they expressed the generalisation error — the difference between E_{out} and E_{in} — in terms of a dataset of size N assumed to be independently and identically distributed (i.i.d.), and a model class $\mathcal{H}$. Here, $\mathcal{H}$ represents a finite set of potential functions that can be fitted to the data. The authors established the following bound:

$$E_{out} < E_{in} + \sqrt{\frac{log(|\mathcal{H}|) + log(\frac{2}{\delta})}{2N}}, \text{ with probability of at least } (1-\delta), \tag{17}$$

where the size of $\mathcal{H}$ (denoted by $|\mathcal{H}|$) is the *complexity* of $\mathcal{H}$, and the error measures taking values in [0,1].

The bound presented in Eq. (17) provides a basis for comparing the out-of-sample performance of local and global forecasting approaches. Under this framework, one method is considered superior to the other if it results in a lower bound on the out-of-sample error. In this context, Montero-Manso and Hyndman (2021) operate under the assumption that

the time series in $\mathbb{S}$ are independent[1]. Their analysis focuses on the average error across all time series $Z_j \in \mathbb{S}$. Consequently, E_{in}^{Local} and E_{out}^{Local} denote the average in-sample and expected out-of-sample errors, respectively, obtained using the local approach. Analogously, E_{in}^{Global} and E_{out}^{Global} represent these errors for the global approach. In this setting, each time series Z_j is modeled using a distinct model class $\mathcal{H}_j$ in the local approach, whereas a single model class $\mathcal{J}$ is employed for the entire set of time series $\mathbb{S}$ in the global approach. Given these considerations, Montero-Manso and Hyndman (2021) propose the following proposition:

PROPOSITION 2—Generalisation bounds for local and global algorithms of finite class: *Local and global approaches have the following generalisation bounds:*

$$E_{out}^{Local} < E_{in}^{Local} + \sqrt{\frac{log(\prod_{j=1}^{J} |\mathcal{H}_j|) + log(\frac{2}{\delta})}{2NJ}}, \quad E_{out}^{Global} < E_{in}^{Global} + \sqrt{\frac{log(|\mathcal{J}|) + log(\frac{2}{\delta})}{2NJ}},$$

with probability of at least $(1 - \delta)$.

PROOF: Let us assume the independence between the time series in $\mathbb{S}$ and that each of these series has a sample of size N. From the Hoeffding inequality Hoeffding (1994) we have: $P(\bar{Y} - \mathbb{E}(\bar{Y}) \geq t) \leq 2e^{-2Nt^2}$, for the independent random variables $Y_1, ..., Y_N$ coming from any probability distribution bounded in [0,1], and where $\bar{Y} = N^{-1} \sum Y_i$, and $t > 0$. If we consider that Y_i is *equivalent* to the in-sample errors produced by one fixed function fitted to a time series of size N, then $\bar{Y}$ will be the average in-sample error and its expectation $\mathbb{E}(\bar{Y})$ will be the expected out-of-sample error. Hence, applying the union bound to the error (Boole's inequality (Pitman, 2012)) when selecting among many functions in $\mathcal{H}$ yields: $P(\bar{Y} - \mathbb{E}(\bar{Y}) \geq t) \leq |\mathcal{H}|2e^{-2Nt^2}$. Moving from one-time series to averaging over all J time series in $\mathbb{S}$, we obtain NJ samples. In the local approach, the size of the class $\mathcal{H}$ is the size of the cartesian product of all $\mathcal{H}_j$ used to fit the time series, whereas, for the global approach, this size is always $|\mathcal{J}|$. Therefore, for the local approach, we have the following: $P(E_{out}^{Local} - E_{in}^{Local} \geq t) \leq \prod_{j=1}^{J} |\mathcal{H}_j|2e^{-2JNt^2}$. If we rewrite the

[1] It is noteworthy that this assumption of independence between time series is not stringent; for a comprehensive discussion, refer to Montero-Manso and Hyndman (2021).

right-hand side of the previous inequality as the tolerance level δ, it follows

$$\prod_{j=1}^{J} |\mathcal{H}_j| 2e^{-2JNt^2} = \delta \Rightarrow t = \sqrt{\frac{log(\prod_{j=1}^{J} |\mathcal{H}_j|) + log(\frac{2}{\delta})}{2NJ}}. \text{ Thus,}$$

$$P\left(E_{out}^{Local} - E_{in}^{Local} \geq \sqrt{\frac{log(\prod_{j=1}^{J} |\mathcal{H}_j|) + log(\frac{2}{\delta})}{2NJ}} \right) \leq \delta$$

$$\Rightarrow P\left(E_{out}^{Local} < E_{in}^{Local} + \sqrt{\frac{log(\prod_{j=1}^{J} |\mathcal{H}_j|) + log(\frac{2}{\delta})}{2NJ}} \right) > 1 - \delta.$$

The proof for the global approach is analogous. *Q.E.D.*

The bounds derived in Proposition 2 allow for a comparison of the generalization capabilities between local and global modeling approaches, even in the absence of prior knowledge regarding the functional class or the data utilized in the modeling process. Suppose we are given a dataset and we aim to identify a global algorithm that either matches or outperforms a given local algorithm in terms of its in-sample error. According to Proposition 2, the comparative performance of the algorithms is entirely contingent upon their respective complexities. Furthermore, Proposition 1 establishes that there exists a global algorithm capable of attaining an in-sample error equivalent to that of a predetermined local method, provided that there is an allowance for a certain level of model complexity. To facilitate a focused analysis on the effect of model complexity, let us equate the in-sample errors of the local and global approaches, yielding $E_{in}^{Local} = E_{in}^{Global}$ and consequently $E_{out}^{Local} = E_{out}^{Global}$. This assumption leads to the equality $\prod_{j=1}^{J} |\mathcal{H}_j| = |\mathcal{J}|$, implying that both algorithms provide the same worst-case guarantees. This scenario indicates that while the complexity of the local approach escalates with the sample size, the complexity of the global approach remains constant. This dynamic yields two significant implications:

First, global algorithms benefit from complex function classes, improving performance. Strategies to enhance complexity involve adding features, nonlinear transformations, or

shifting from linear to neural network models. This shift is supported by their success in forecasting competitions like M4 and M5 Makridakis et al. (2020), Spiliotis et al. (2022) and empirical evidence Montero-Manso and Hyndman (2021) showing deep network models outperform linear ones across datasets. Literature Bandara et al. (2020c, 2019, 2020b), Hewamalage et al. (2021a) confirms global models' performance over local algorithms using the same datasets.

Second, as the sample size NJ swells, the complexity of the local approach, $\prod_{j=1}^{J} |\mathcal{H}_j|$, correspondingly increases, whereas the complexity of the global approach, $|\mathcal{J}|$, remains invariant. Under these conditions, and maintaining equivalent in-sample losses ($E_{in}^{Local} = E_{in}^{Global}$), the upper bound on the out-of-sample error for the global model (E_{out}^{Global}) will be consistently lower than that of the local model (E_{out}^{Local}), signifying enhanced generalization capabilities for global models. This insight proves instrumental in extrapolating the consistency established for local approaches to their global counterparts, a topic explored in the subsequent section.

Consis

3.2. *Consistency of Smooth Local Feedforward Neural Network Estimators*

In this section, we establish the consistency of smooth local Multi-Layer Perceptrons (MLPs), drawing upon the foundational work of Fallahgoul et al. (2024) and related contributions from Farrell et al. (2021), Fabozzi et al. (2019), Horel and Giesecke (2020). We commence by delineating the regression framework employed by Deep Neural Network (DNN) models, as well as clarifying the structure of the target function. Subsequently, we describe the configuration of DNNs. In the final part of this section, we demonstrate how local Feedforward Neural Network (FFNN) estimators fulfill the consistency criterion.

3.2.1. *The regression framework and structure of target function*

Let us consider a fixed probability space $\left(\Omega, \mathcal{F}, \mathcal{P}\right)$ where the target Y is related to exogenous variables $\mathbf{X} \in \mathbb{X} := \left[-1, 1\right]^d$ in a regression modelling specification such as $Y = f_*\left(\mathbf{X}\right) + \epsilon$. Fallahgoul et al. (2024) need to state a first assumption as follows.

ASSUMPTION 7: *Assume a target variable Y whose absolute value is superiorly bounded by a large arbitrary parameter $M_Y > 0$, a covariate vector $\boldsymbol{X}$ continuously dis-*

tributed according to $\mathcal{P}$, and the joint law of both $\boldsymbol{X}$ and ϵ, which is absolutely continuous w.r.t. Lebesgue measure Δ. Thus, concerning the error variable ϵ, we assume ϵ is independent of $\boldsymbol{X}$, $\mathbb{E}[\epsilon] = 0$, $|\epsilon| \leq 1$, and $\mathbb{E}[\epsilon^2] = \sigma^2 < \infty$.

Considering f_* is the true mean of Y, we also may assume that

$$\left\|f_*\right\|_\infty + \sum_{\substack{\overrightarrow{\alpha} \in (\mathbb{N} \cup \{0\})^d \\ \|\overrightarrow{\alpha}\|_1 = 1}} \left\|\nabla^{\overrightarrow{\alpha}} f_*\right\|_\infty \leq 2M, \tag{18}$$

where $\overrightarrow{\alpha}$ stands for a multi-index notation. Thereby, the upper bounds of f_* and ϵ imply that $|Y| < 2M + 1$. To guarantee ϵ is bounded, Fallahgoul et al. (2024) normalize Y dividing it by the constant M_Y, which generates a new bounded error term, $-1 \leq \frac{\epsilon}{M_Y} \leq 1$, such that the regression model can be rewritten as

$$\left|\frac{Y}{M_Y}\right| = \left|\frac{f_*(\mathbf{X}) + \epsilon}{\epsilon}\right| \leq \left|\frac{f_*(\mathbf{X})}{M_Y}\right| + \left|\frac{\epsilon}{M_Y}\right| \leq 1. \tag{19}$$

Beyond Assumption 7, Fallahgoul et al. (2024) establish two more assumptions related to the target function f_* structure.

ASSUMPTION 8: *Assume the target function $f_* \in \mathcal{C}^{\lfloor d/2 \rfloor + 2}\Big(\big[-1,1\big]^d\Big) \cap \mathbb{H}^{\lfloor d/2 \rfloor + 2,2}\Big(\big[-1,1\big]^d\Big)$, where $\mathbb{H}^{\lfloor d/2 \rfloor + 2,2}\Big(\big[-1,1\big]^d\Big)$ indicates the Sobolev space with weak differentiation order equal to $\lfloor d/2 \rfloor + 2$ and the function's weak derivatives in this space is in $L^2\big([-1,1]^d\big)$.*

ASSUMPTION 9: *Assume the target function f_* has a binary tree structure with a certain depth and quantity of nodes as stated in Poggio et al. (2017).*

Assumption 8 requires that the function f_* belongs to the Sobolev space $\mathbb{H}^{\lfloor d/2 \rfloor + 2,2}\Big(\big[-1,1\big]^d\Big)$, which ensures that the integral of the entropy number of the function class $\mathcal{F}$ can be bounded (the construction of $\mathcal{F}$ is detailed in Section 3.2.2 that follows). Additionally, f_* must also be an element of the function space $\mathcal{C}^{\lfloor d/2 \rfloor + 2}\Big(\big[-1,1\big]^d\Big)$, ensuring its differentiability. According to Fallahgoul et al. (2024), the order of differentiation for f_* must be at least $\lfloor d/2 \rfloor + 2$, a condition necessary to apply Corollary 4 from Nickl and Pötscher (2007) in their proofs.

Furthermore, Assumption 9 is vital for the application of Theorems 1 and 2 from Poggio et al. (2017). For a comprehensive discussion on the rationale behind the assumption of a binary tree visual structure for the target function f_*, readers are referred to Fallahgoul et al. (2024). It is crucial to note that this binary tree structure is only required for the target function f_* and not for the least squares estimator $\hat{f}_n$, which is derived as a Deep Neural Network (DNN) following Eq. (24). Additionally, it is assumed that both the complexity of the target function (represented by the number of nodes, N_f, in the binary tree visualization of f_*) and its depth, L_d, remain constant. Fallahgoul et al. (2024) argue that this assumption is not overly restrictive since the depth of a binary tree structure representing a function can always be increased. As long as the depth L_d is sufficiently large, it ensures that the number of nodes N_f will also be adequately large, as discussed in Poggio et al. (2017).

3.2.2. *DNN setting*

NNs can be designed with a variety of architectures or classes, collectively denoted as $\mathcal{F}_{DNN}$. In this context, we direct our attention to the class of FFNNs, characterized by specific network components. These components include an input layer consisting of d units (corresponding to the covariates $\mathbf{X} \in \mathbb{R}^d$), a single output layer for the target outcome Y, and L hidden layers. The parameter L signifies the network's depth, i.e., the number of hidden layers. A depth of one indicates a shallow network, while a depth greater than one corresponds to a deep network. Importantly, we index L with the input size d, denoting it as L_d, to highlight its independence from the sample size n and potential dependence on d. The input size d influences the required network depth, as larger values of d may necessitate a deeper network. Consequently, d serves as a crucial factor in determining the depth L_d of a NN. The hidden layers of the network comprise one or more nodes, with the number of nodes in each layer denoted by $H_l = H_n$ (where $l = 1, 2, ..., L_d$). Here, H_n represents the network's width and is indexed by n to indicate its dependency on the sample size. An activation function $\psi(\cdot) : \mathbb{R} \to \mathbb{R}$ is applied across all hidden nodes. In FFNNs, these nodes are interconnected through a directed acyclic graph, ensuring unidirectional information flow. In this setting, we assume a fully connected FFNN architecture with a consistent number of hidden nodes (H_n) across all layers, and a single node in the output layer, yielding a single estimated value for the target. This configuration aligns with a specific subclass of Multi-Layer Perceptrons (MLPs; refer to Fig. A.7 in A.7 of **Online Appendix** for a visual

representation). It is important to note, as demonstrated by Farrell et al. (2021) in their Lemma 1, that any FFNN can be transformed into an MLP, ensuring that this assumption does not constrain the class of FFNNs under consideration. Given these parameters, the network's complexity is denoted as $\mathcal{U} = \mathcal{O}(H_n L_d)$. With these definitions in place, we can now proceed to construct a family of MLPs, characterized by depth L_d and width H_n as follows:

$$\mathcal{F}_{DNN,H_n} = \{o_{L_d+1,1}(\mathbf{X}) | \mathbf{X} \in \mathbb{X}\}, \tag{20}$$

where $o_{l,u}(\mathbf{X})$ is the NN's output of the u^{th} node of layer l which takes $\mathbf{X}$ as input, $l = L_d + 1$ corresponds to the output layer (while $l = 0$ is the input layer), with $1 \leq l \leq L_d$ and $u \in \{1, ..., H_{n,l}\}$, where $H_{n,l}$ is the amount of hidden nodes for the l^{th} hidden layer. The formula for $o_{l,u}(\mathbf{X})$ is

$$o_{l,u}(\mathbf{X}) = \psi\left(\sum_{k=1}^{H_{n,l-1}} \gamma_{l,u,k} \cdot o_{l-1,k}(\mathbf{X}) + \gamma_{l,u,0}\right), \tag{21}$$

where $o_{0,k}(\mathbf{X}) = x_k$, the k^{th} element of $\mathbf{X}$, $\gamma_{l,u,k}$ and $\gamma_{l,u,0}$ are respectively the weight and bias parameters for the node (l, u), and the activation function defined as $\psi : \mathbb{R} \to \big[-b, b\big]$, with $b > 1$ and where $\psi \in \mathcal{C}^\infty$ is a smooth and non-polynomial function (for our FFNN's implementation, the $\psi \equiv tanh(\cdot)$ function with range (-1,1)). Moreover, to execute the proof of consistency for the FFNN estimator, we need differentiable activation functions to apply Theorem 2 of Poggio et al. (2017), which requires the NN's infinite differentiability with respect to each weight parameter. Furthermore, the larger range of $tanh$ may lead to better performance than $sigmoid$ activation functions, for instance.

For the output layer, i.e., $l = L_d + 1$, $u = 1$ because the output layer has only one single node. Also, in the output layer, the activation function is not applied to the output node. Hence, Eq. (21) for the output layer is defined as

$$o_{L_d+1,1}(\mathbf{X}) = \sum_{k=1}^{H_{n,L_d}} \gamma_{L_d+1,1,k} \cdot o_{L_d,k}(\mathbf{X}) + \gamma_{L_d+1,1,0} \tag{22}$$

or equivalently, in terms of a DNN function $f_{DNN,H_n} \in \mathcal{F}_{DNN,H_n}$ as

$$f_{DNN,H_n}(\mathbf{X}) = \sum_{k=1}^{L_d} \gamma_{L_d+1,1,k} \cdot o_{L_d,k}(\mathbf{X}) + \gamma_{L_d+1,1,0} = \gamma_{L_d+1} \cdot \mathbf{o}_{L_d,k}(\mathbf{X}) + \gamma_{L_d+1,1,0} \tag{23}$$

Fallahgoul et al. (2024) assume $||\gamma_{l,u,k}||_1 \leq Mb$, where $M \in \mathbb{R}^+$ is an arbitrary parameter like in Eq. (18), making $\mathcal{F}_{DNN,H_n}$ the function class of all bounded-weight DNN functions with fixed-width H_n and fixed-depth L_d. This assumption is essential for using Theorem 14.5 in Anthony and Bartlett (2009) that requires this weight bound. Finally, we apply a least square estimator $\hat{f}_n \in \mathcal{F}_{DNN,H_n}$ that is obtained by minimizing the loss function $l(\cdot)$ as follows:

$$\hat{f}_n = \underset{f_n \in \mathcal{F}_{DNN,H_n}}{\arg\min} \sum_{i=1}^{n} l\big(f, \mathbf{x}_i, y_i\big), \text{with} \tag{24}$$

$$l\big(f, \mathbf{x}_i, y_i\big) := \frac{1}{2}\big(y_i - f(\mathbf{x}_i)\big)^2 = \frac{1}{2}\big(f_*(\mathbf{x}_i) + \epsilon_i - f(\mathbf{x}_i)\big)^2 \tag{25}$$

where $\big(\mathbf{x}_i, y_i\big)$ are samples from the total n observations of $\big(\mathbf{X}, Y\big)$.

3.2.3. *Consistency of smooth FFNN*

This section shows that $\hat{f}_n$ in DNN form as in Eq. (24) is a consistent estimator of f_*, i.e., $\hat{f}_n$ is approximately equal to f_* as $n \to \infty$. Farrell et al. (2021) present notable findings on the subject of DNN inference, including nonasymptotic bounds for nonparametric estimation utilizing various NN architectures. Their key innovation lies in localization and the avoidance of assuming bounded weights for multi-layer MLP. By relaxing this assumption, their work makes a substantial contribution to the existing literature. Although Fallahgoul et al. (2024) adopt a similar strategy to establish the consistency of $\hat{f}_n$, unlike Farrell et al. (2021), they prove the consistency for **smooth** MLPs (instead of MLPs using the ReLU activation function), which matches the FFNN architecture we implement in this article. The choice of smooth MLPs is made because of the necessity of their parallel containment[2] to build a sieve space, which Fallahgoul et al. (2024) use to derive the asymptotic distribution of the smooth MLP estimator. They start considering some definitions of the DNN's approximation error ϵ_n of f_* by f_n. The definitions of the $\|\cdot\|_\infty$-minimiser of $\mathcal{F}_{DNN,H_n}$ and the related $\|\cdot\|_\infty$ distance to f_* are

$$f_n := \underset{f_n \in \mathcal{F}_{DNN,H_n}}{\arg\min} \big\|f - f_*\big\|_\infty, \text{ and } \epsilon_n := \big\|f_n - f_*\big\|_\infty \tag{26}$$

[2]Parallel containment stands for any two smooth MLPs with identical depth that can be combined to obtain a new single smooth MLP.

It is important to note that f_n in Eq. (26) and $\hat{f}_n$ in Eq. (24) are different because the former represents the optimal $\|\cdot\|_\infty$-approximator whereas the latter is the least squares approximator. The consistency of the least squares estimation is ensured by the existence of an upper bound of order $o_P(1)$ for both $\|\cdot\|_{L^2(P)}$ and the semi-norm $\rho_n(g(\mathbf{X})) := \sqrt{\frac{1}{n}\sum_{i=1}^n (g(\mathbf{x}_i))^2}$, where $\mathbf{x}_i$ are empirical samples of $\mathbf{X}$. The proof of this statement is found in Appendix A of Fallahgoul et al. (2024). Additionally, ϵ_n demands an upper bound as stated in the following Theorem 1. To prove this upper bound of ϵ_n, Fallahgoul et al. (2024) use the results of Poggio et al. (2017) for NNs with smooth and non-polynomial activation functions.[3] To be able to use the results of Poggio et al. (2017), the following additional assumption is necessary.

ASSUMPTION 10: *The boundaries imposed for the weight parameters of $\mathcal{F}_{DNN,H_n}$ ($\gamma_{l,u,k}$) do not restrain us from employing Theorems 1 and 2 of Poggio et al. (2017).*

Fallahgoul et al. (2024) discuss that Assumption 10 is not restrictive since the weight boundaries of NNs that belong to $\mathcal{F}_{DNN,H_n}$ depend only on M which can be made arbitrarily large. Thereby, under the Assumptions 7, 8, 9, and 10 we can achieve the following main result.

THEOREM 1—Consistency and Upper Bound of Estimation: *Let $\hat{f}_n$ be the least squares estimator as defined by Eq. (24), for a loss function observing Eq. (25), with fixed depth L_d and fixed width H_n. Thus, with the probability of at least $1 - e^{-p}$, for large enough n,*

$$\|\hat{f}_n - f_*\|_{L^2(P)} \leq \mathcal{U}(C', \epsilon_n) := C'\left(\sqrt{\frac{H_n^2 L_d^2 ln(H_n L_d n)}{n}} + \sqrt{\frac{ln(ln(n)) + p}{n}} + \epsilon_n\right),$$

where the order of $p = p(n)$ must be less than $H_n^2 L_d^2 ln(H_n L_d n)$ and the constant $C' > 0$ must be independent of n, but may depend on d, M, and other fixed constants. In addition, the same upper bound applies for $\rho_n((\hat{f}_n - f_*)(\mathbf{X}))$.

The detailed proof of Theorem 1 is provided in Appendix A of Fallahgoul et al. (2024). We can consider Theorem 1 as an extension of the results obtained in Theorem 2 of Farrell

[3] Distinctly from Farrell et al. (2021) who prove this only for ReLU networks by employing Theorem 1 of Yarotsky (2017).

et al. (2021) for smooth NNs. It allows us to control the upper bound $\mathcal{U}(C', \epsilon_n)$ by changing the number of hidden nodes H_n in each hidden layer for networks contained in $\mathcal{F}_{DNN,H_n}$. To control the convergence rate we only need to adjust the values of p since $p = p(n) = order\big(H_n^2 L_d^2 ln\big(H_n L_d n\big)\big)$, which implies

$$\mathcal{U}(C', \epsilon_n) = \mathcal{O}\Big(n^{-1}\Big(H_n^2 L_d^2 ln\big(H_n L_d n\big) + ln\big(ln(n)\big)\Big)\Big) \to 0, \text{ as } n \to \infty.$$

3.3. *The consistency of smooth* global *FFNN estimators*

In this section, we extend the consistency results achieved in Theorem 1 to GFM-FFNN estimators by combining the results of Preposition 2 with Theorem 1.

In Theorem 1 we obtain the consistency for *local* FFNNs belonging to $\mathcal{F}_{DNN,H_n}$ by the existence of an upper bound $\mathcal{U}(C', \epsilon_n)$ for ϵ_n which converges to zero for large enough sample sizes n. In the other hand, by Preposition 2, in Section 3.1.2, we see that the complexity of the local approach, $\prod_{j=1}^{J} |\mathcal{H}_j|$ increases with the sample size n, while the global complexity, $|\mathcal{J}|$, remains constant. That implies, for same in-sample losses $E_{in}^{Local-FFNN} = E_{in}^{Global-FFNN}$, that

$$\begin{aligned} E_{out}^{Global-FFNN} &< E_{in}^{Global-FFNN} + \sqrt{\frac{log(|\mathcal{J}|) + log(\frac{2}{\delta})}{2nJ}} \\ &< E_{in}^{Local-FFNN} + \sqrt{\frac{log(\prod_{j=1}^{J} |\mathcal{H}_j|) + log(\frac{2}{\delta})}{2nJ}} \leq \mathcal{U}(C', \epsilon_n) \to 0, \text{ as } n \to \infty, \end{aligned}$$

which means for large enough n, the bound of out-of-sample errors for a global model (for example, a global FFNN), $E_{out}^{Global-FFNN}$, becomes smaller than the bound of out-of-sample errors for local models, $E_{out}^{Local-FFNN}$, which in turn, if consistent, are bounded by $\mathcal{U}(C', \epsilon_n)$. Therefore, when the local version of an FFNN estimator is consistent (in terms of Theorem 1), this implies that by Proposition 2, the global-FFNN estimator also is consistent. Hence, as a corollary to Theorem 1, we obtain the consistency of *global* FFNN estimators.

COROLLARY 1.1—Consistency of *global* FFNN estimators: *If* $\hat{f}_n$ *in* local-*FFNN form belonging to* $\mathcal{F}_{DNN,H_n}$ *as in Eq.* (24) *is a consistent estimator of* f_* *in terms of Theorem 1, then the* global *version of this same FFNN also is a consistent estimator.*

According to Definition 1 in Carvalho et al. (2018), the estimation of the ATE, as such as in our work, is given by $\hat{\Delta}_T = \frac{1}{T-T_0}\sum_{t=T_0+1}^{T}\hat{\delta}_t$, where $\hat{\delta}_t \equiv \mathbf{y}_t - \mathcal{M}(Y^{(0)}_{j,t\leq T_0}, X^{(0)}_{j,t\leq T_0}, \hat{\theta})$ for $t = T_0+1, ..., T$ and $\hat{\theta}$ is a consistent estimator for the true θ using only the first T_0 observations of pre-intervention data. In their work, Carvalho et al. (2018) provide and prove Theorem 1 and Lemma 1 to demonstrate that any statistical model, such as $\mathcal{M}(\cdot)$, that satisfies a set of assumptions will yield consistent and asymptotically normal estimates for the ATE defined as their Definition 1. Therefore, the ATE estimations carried out by our framework using *global* FFNN estimators can be considered consistent.

FFNNs struggle with temporal dependencies due to a lack of internal memory, unlike LSTMs, which excel in learning long-term dependencies and often outperform FFNNs in time series forecasting. This trend is supported by experimental evidence, such as Peng et al. (2022)'s findings on LSTM's reduced training errors with larger data. However, LSTM's complexity challenges formal consistency proof. While no definitive proof exists, Fallahgoul et al. (2024) suggests establishing LSTM consistency as a research direction. Our experiments also show DeepCPNet-LSTM's superiority over DeepCPNet-FFNN without a formal consistency proof. See Table A.IV in the online appendix for a detailed comparison.

4. EXPERIMENTS

4.1. *Simulation study*

Given the unobservable nature of counterfactuals, it's challenging to gauge how close estimates are to the true treatment effect. Synthetic data and computer simulations are typically used to identify estimates nearer to the actual effect. This method allows for the recreation of the true scenario without intervention, facilitating a comparison between the model-predicted counterfactual and treatment effect. We follow the experimental simulation approach in Grecov et al. (2022) and simulate from a VAR(4) macro model of the US economy. We use the growth rate of government expenditures (RGCE) as the intervention variable and investigate the impact on the unemployment rate (UNRATE) given the growth rate of the Consumer Price Index (RCPI), and the growth rate of the Gross Domestic Product (RGDP). We decided to use a macro model as it allows us sense check potential causal effects in terms of their magnitudes. We utilize Monte Carlo simulation of the VAR system. As a result, the perturbations in the RGCE variable, which represent our intervention, man-

ifest in the RCPI, RGDP, and UNRATE time series, creating a complex interplay among these variables. This is visually represented in Fig. A.10 in A.7 of **Online Appendix**.

We explored two scenarios: one with interventions affecting only treated units (where the results are explored in the **Online Appendix**, Section A.5), and another with spillover effects on control units (results discussed in Section 5.1). Our goal was to compare the efficacy of ATE estimation methods using pre-treatment data versus those incorporating both pre- and post-intervention data.

In the first scenario, where spillover effects are absent, we introduced a level shift varying randomly between 3.5% and 4%. This shift was applied to 30 of the 100 synthetic time series, designated as our treated units. The remaining 70 series, unaffected by the intervention, served as control units. The intervention spanned from $T_0 = 220$ to the series' end at 224, yielding an in-sample (pre-treatment) data length of 219 and a forecasting horizon (post-treatment period) of 25. In the second scenario, which incorporates spillover effects, a smaller level shift ranging from 1.75% to 2% was applied to 35 series, previously categorized as control units in the first scenario. These series now represent our control instances subject to spillover effects. The remaining 35 series continue to serve as control units without spillovers.

This simulation aims to compare the DeepCPNet and benchmark models in estimating the impact of government expenditure increases (RGCE) on unemployment (UNRATE) under various scenarios, including spillover effects. We evaluate model performance using three metrics: i) Point forecast accuracy via RMSE, WRMSPE, and sMAPE; ii) ATE estimation precision with RMSE and WRMSPE; iii) Statistical power, the likelihood of correctly rejecting the null hypothesis of no treatment effect. These indicators provide a holistic view of model effectiveness in detecting effects and forecasting accuracy. Formulas and more details about these error metrics are found in the **Online Appendix** A.4.

4.2. *The real-world datasets and design of empirical experiment*

We utilized the Nielsen household-level panel dataset (NielsenIQ-Consumer Panel) from the Kilts Center at the University of Chicago Booth School of Business, featuring weekly

U.S. household grocery purchases from January 2016 to December 2018.[4] The dataset records include grocery items bought, item prices, purchase dates, deal flags, total spend per trip, store information, and demographic details. We enriched this dataset with Nielsen's retail scanner data (NielsenIQ-Retail Scanner[4]) to track weekly store prices and identify discount offerings—marked by at least a five percent price drop.

Following Mojir and Sudhir (2021), we choose milk to study deal effects on store sales, highlighting its role as a loss leader. Milk's regular purchase pattern and perishability make it an ideal case, with retail data showing it has the second-highest basket share and the highest penetration rate of 88% among major categories (Mojir and Sudhir (2021)).

We constructed a dataset from shopping trips, including milk, focusing on stores with annual purchases over at least 47 weeks. Using Nielsen data, we identified a treatment period from December 1-29, 2018, for promotional analysis, resulting in 152 weeks of data and a 2-week forecast horizon. Stores with promotions were treated units; others were controls, totalling 148 stores (102 treated, 46 control). To examine market overlap and spillover effects, we analyzed sales by store zip code, categorizing control stores by proximity to treated ones. The analysis used post-treatment data against a pre-treatment framework to assess spillover impacts on sales.

Utilizing Nielsen's and U.S. holidays (2016-2018) datasets, we accounted for holiday sales fluctuations. This data, comprising holiday names and types (national, religious, cultural, sporting), was integrated as dummy variables in our econometric model to predict "sales". Details are in Table A.V in the **Online Appendix**.

5. RESULTS AND DISCUSSION

In this section, we benchmark the results of our proposed framework against some well-established univariate and *local* time series forecasting models and some structural causal models. For an overview of the benchmark models used in all experiments, see Table A.I as well as the implementation details in the **Online Appendix** A.4.

[4]This analysis partly uses Nielsen Consumer LLC data and NielsenIQ Datasets at the Kilts Center for Marketing Data Center at The University of Chicago Booth School of Business. Interpretations from NielsenIQ data are the researchers' and do not reflect NielsenIQ's views, which bears no responsibility for the analysis or results.

5.1. *The simulated datasets with spillover effects*

In simulations with nonlinear time series, heterogeneous interventions, and spillover effects, DeepCPNet outperforms benchmarks (Table I). The synthetic intervention allows precise effect measurement on treated units and evaluation on two control groups: those affected by spillovers ("Control Units - spillover") and those unaffected ("Control Units").

In Table I, DeepCPNet exhibited lower error rates in outcome forecasting and ATE estimation for treated units, effectively identifying null effects in unaffected control units. This demonstrates its enhanced performance in scenarios where the SUTVA assumption is violated, in contrast to most benchmarks, except for DeepAR, which inaccurately incorporated post-treatment data and external predictors influenced by peer treatment interference. ArCo, while effective without spillovers, struggled with ATT and spillover effect predictions due to interference. Statistical significance testing, outlined in Section 2.3.2, confirmed significant effects with distinct accuracy levels across models, as indicated by p-values for the difference in means test (Table I, last column), and Fig. A.5 visually illustrates model proximity to true effects.

Control units with spillover effects' paths matched benchmark model estimates from post-treatment data, as Fig. A.5 (panel B) shows. DeepCPNet, using only pre-treatment data, uniquely forecasted outcomes close to true counterfactuals. It alone confirmed the null effect in placebo tests for spillover-free control units, with a p-value > 0.01, depicted by the concentrated density curve around zero in Fig. 1.

Assessing treatment effects' significance involves examining estimated ATE confidence intervals (CIs). Non-significant effects have CIs including the null effect, while significant effects' CIs exclude it. This study used quantile regression for probabilistic forecasting (Grecov et al. (2022)) and bootstrapping (100 repetitions) for CI construction. Bootstrapping's limitation is its narrower intervals, reflecting only parameter uncertainty, unlike probabilistic forecasting, which also considers model and data uncertainty. The findings, shown in Table II for suitable models, align with those in Table I.

The experiment confirms how the spillover effects biased the counterfactual generation for the treated units by the models that employ post-treatment data for ATT and ATE_{ctr} estimations.

TABLE I

AVERAGE AND CUMULATIVE TREATMENT EFFECT ESTIMATES AND THEIR ACCURACY FOR TREATED AND CONTROL UNITS POST-INTERVENTION - SIMULATION CONSIDERING THE OCCURRENCE OF SPILLOVERS.

	MODEL	Average Effect (ATT/ATE$_{ctr}$)[a]	Total Effect[b]	RMSE (for ATT/ATE$_{ctr}$)	Mean RMSE (outcome point forecasting vs. true counterfactual)	Mean WRMSPE (outcome point forecasting vs. true counterfactual)	Mean sMAPE (outcome point forecasting vs. true counterfactual)	p-values ($H_0 : ATT/ATE_{ctr} = 0$)[c]	p-values ($H_m : \bar{e}_0 = \bar{e}_m$)[d]
Treated Units	*(True Effect)*	*-1.890*	*-47.25*	-	-	-	-	-	-
	DeepCPNet-LSTM	-1.859	-46.47	**0.0313**	**0.0788**	**0.0129**	**0.0104**	1.12e-69	-
	DeepCPNet-FFNN	-1.941	-48.53	0.0513	0.0799	0.0130	0.0117	2.39e-85	0.9990
	DeepAR	-1.768	-44.20	0.1218	0.1485	0.0242	0.0203	5.17e-107	0.9448
	Global-GAM	-1.098	-27.46	0.7916	0.8783	0.1433	0.1401	5.86e-81	6.0e-12
	CausalImpact	-1.510	-37.75	0.3798	0.4220	0.0689	0.0643	7.02e-100	0.0051
	ArCo	-1.470	-36.76	0.4196	0.4524	0.0738	0.0711	7.41e-104	8.8e-08
	DiD	-1.462	-	0.4283	-	-	-	6.03e-70	-
Control Units - spillover	*(Expected Effect)*	*0*	*0*	-	-	-	-	-	-
	(True Effect/Spillover)	*-0.382*	*-9.54*	-	-	-	-	-	
	DeepCPNet-LSTM	-0.423	-10.58	**0.0415**	**0.1095**	**0.0185**	**0.0137**	3.32e-16	-
	DeepCPNet-FFNN	-0.465	-11.63	0.0836	0.1249	0.0212	0.0188	7.57e-19	0.5564
	DeepAR	0.039	0.98	0.4209	0.4454	0.0754	0.0770	0.25200	1.5e-10
	Global-GAM	-0.243	-6.08	0.1384	0.1499	0.0254	0.0251	3.40e-16	0.1768
	CausalImpact	-0.245	-6.12	0.1368	0.2113	0.0358	0.0306	0.0000159	0.0866
	ArCo	-0.071	-1.77	0.3108	0.3459	0.0586	0.0563	0.0000196	1.2e-06
	DiD	-0.538	-	0.1565	-	-	-	1.34e-10	-
Control Units	*(True Effect)*	*0*	*0*	-	-	-	-	-	-
	DeepCPNet-LSTM	0.225	5.63	**0.2251**	**0.2605**	**0.0444**	**0.0394**	0.26200	-
	DeepCPNet-FFNN	0.249	6.22	0.2488	0.2867	0.0489	0.0435	0.04320	0.456
	DeepAR	0.334	8.35	0.3339	0.3669	0.0625	0.0586	6.60e-21	0.456
	Global-GAM	0.288	7.20	0.2879	0.3114	0.0531	0.0501	3.55e-09	0.832
	CausalImpact	0.400	10.01	0.4004	0.4708	0.0802	0.0762	0.00000331	8.4e-05
	ArCo	0.353	8.83	0.3532	0.3921	0.0668	0.0619	0.	0.024
	DiD	0.538	-	0.5381	-	-	-	1.34e-10	-

[a] Values are calculated according to Eq. (**??**), i.e., averaged for all treated units and through the full intervention period.

[b] These are the cumulative sum of treatment effects for all units through the whole intervention period.

[c] Zero Effect Null Hypothesis Test for units according to statistics described in Section 2.3.2 (Chernozhukov et al. (2021)) and Eq. (14), except for DiD obtained throughout the `lm()` regression with robust standard errors.

[d] Statistical two-tailed test for significance of the differences between point forecasting mean errors of DeepCPNet-LSTM and each one of the $m = 1, ..., 8$ benchmark models for $\alpha = 0.05$.

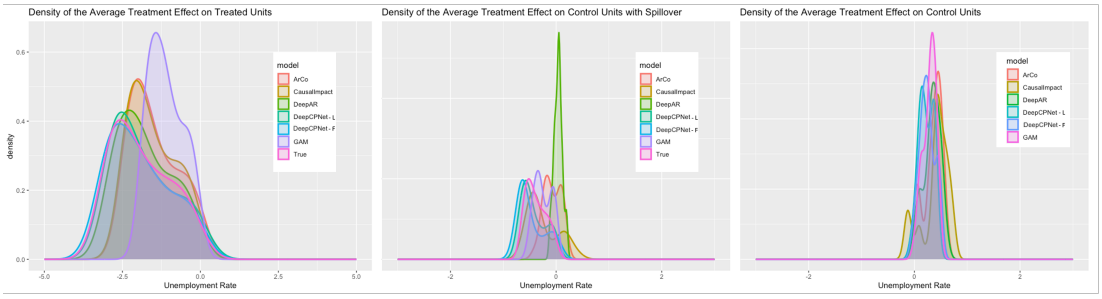


FIGURE 1.—Density distribution of the ATT/ATE$_{ctr}$ per model for each group of units.

5.2. *The effect of discount and deal offers on the retail sales for US*

Considering the competitive retail market, our study estimates the impact of deals on essential products on grocery store sales. We posit that deals in select stores may cause con-

TABLE II

95% CONFIDENCE INTERVALS FOR AVERAGE TREATMENT EFFECT ESTIMATES FOR TREATED AND CONTROL UNITS POST-INTERVENTION - SIMULATION CONSIDERING THE OCCURRENCE OF SPILLOVERS.

	MODEL	Average Effect (ATT/ATE_{ctr})	ATE 95% Confidence Interval Probabilistic Quantile Regression	Bootstrapping
Treated Units	DeepCPNet-LSTM	-1.859	[-2.206, -1.443]	[-2.119, -1.576]
	DeepCPNet-FFNN	-1.941	[-2.212, -1.505]	[-2.152, -1.730]
	DeepAR	-1.768	[-2.765, -0.945]	[-2.881, -0.655]
Control Units - spillover	DeepCPNet-LSTM	-0.423	[-0.721, -0.008]	[-0.623, -0.135]
	DeepCPNet-FFNN	-0.465	[-0.749, -0.015]	[-0.673, -0.257]
	DeepAR	0.039	[-1.260, 0.469]	[-1.025, 1.104]
Control Units	DeepCPNet-LSTM	0.225	[-0.106, 0.610]	[-0.002, 0.490]
	DeepCPNet-FFNN	0.249	[-0.033, 0.643]	[0.021, 0.427]
	DeepAR	0.334	[-0.801, 1.227]	[-0.957, 1.574]

sumption shifts and spillover effects on competitors without similar promotions. Our approach aims to provide precise ATT estimations, addressing spillovers and possible SUTVA assumption breaches.

The causal effect estimation of milk deals on store sales obtained from DeepCPNet is compared to the traditional econometric causal models CausalImpact, ArCo and DiD.

In the CSC method, true impacts on treated entities are unobservable. Thus, we assess model accuracy using control unit errors—stores without milk deals. This yields higher errors for treated units (Table III), indicating that milk deals likely to boost sales. Notably, control stores near treated ones showed similar or greater errors, hinting at spillover effects from promotions. Such consumer shifts towards stores with better deals, implying competitive influence, will be detailed in ATE estimation discussions.

Given the likelihood of spillover effects affecting some control stores, the point forecasting performance is more accurately assessed using the error metrics for the “control-out" series. In this regard, DeepCPNet surpasses all benchmark models, demonstrating superior performance in terms of RMSE and WRMSPE error metrics, as shown in Table III.

Comparing forecasts to actual data (Fig. A.6), the first week shows a significant impact on treated stores, aligning with spillover effects in nearby control stores (plot A). Plot (B) reveals a sales drop in these control stores initially. Control stores outside the treated area show minimal impact (plot C). Our analysis thus concentrates on the first two weeks’ impact of milk deals on sales, elaborated in Table III’s last four columns.

TABLE III

RESULTS FOR THE POINT FORECASTING BY GROUP OF STORES AND THE CAUSAL EFFECTS FOR THE FIRST TWO WEEKS OF TREATMENT BY GROUP OF STORES - REAL-WORLD APPLICATION.

	MODEL	RMSE (for point forecasting)	WRMSPE (for point forecasting)	Average Effect (ATT/ATE_{ctr}) Week 1	Average Effect (ATT/ATE_{ctr}) 2 Weeks	*p-values*[a] Week 1	*p-values*[a] 2 Weeks
Treated Units	DeepCPNet	39.463	0.153	40.88 *(+14.5%)*	13.37 *(+4.9%)*	7.3e-07	7.0e-04
	CausalImpact	25.326	0.092	-30.61 *(-10.8%)*	-23.67 *(-8.7%)*	0.2470	0.2320
	ArCo	36.032	0.131	-13.15 *(-4.7%)*	-24.20 *(-8.9%)*	0.4704	0.3876
	DiD	-	-	24.09 *(+8.6%)*	25.99 *(+10.1%)*	0.4210	0.2356
Control Units - IN[b]	DeepCPNet	35.513	0.147	-39.09 *(-19.6%)*	-40.27 *(-19.1%)*	0.5059	0.2951
	CausalImpact	55.329	0.230	-101.13 *(-50.7%)*	-70.59 *(-33.5%)*	0.0252	0.0247
	ArCo	69.493	0.288	-100.73 *(-50.5%)*	-88.20 *(-42.8%)*	0.0045	0.0004
	DiD	-	-	-103.83 *(-53.1%)*	-59.05 *(-28.2%)*	0.0168	0.0762
Control Units - OUT[c]	DeepCPNet	**28.581**	**0.145**	-9.63 *(-4.3%)*	-23.76 *(-12.5%)*	0.9359	0.3870
	CausalImpact	49.998	0.256	-29.21 *(-12.9%)*	-52.54 *(-27.5%)*	0.2882	0.2368
	ArCo	37.081	0.190	0.43 *(+0.2%)*	-31.92 *(-16.7%)*	0.9329	0.1154
	DiD	-	-	103.83 *(+46.9%)*	59.05 *(+31.1%)*	0.0168	0.0762

[a] *p-values* refers to the zero effect test for the null hypothesis, i.e., $H_0 : ATT/ATE_{ctr} = 0$.
[b] Control Units - IN stands for the control stores localised in the same region as the treated stores.
[c] Control Units - OUT stands for the control stores outside the treated region.

In Table III, benchmarks missed sales growth in stores offering milk deals, unlike DeepCPNet, which detected a 15% sales increase in the first week, supported by its p-values. Benchmarks struggled due to using post-intervention data from spillover-affected controls, skewing results for unaffected control units. DeepCPNet accurately captured the null effect for these "control-out" units with higher p-values, a difference highlighted in Fig. 2. Plot (B) shows DeepCPNet's precise reflection of the positive effect in treated stores' first week, while plot (C) indicates the spillover effect's impact on "control-in" stores, contrasted with the minimal deviation in "control-out" stores (plot A).

In section A.6 in the online appendix, we discuss the estimation of CATE by splitting the sample.

6. CONCLUSIONS

While numerous methods assess policy effects, identifying treatment effects is hard, especially with spillovers beyond the primary target. This limits baseline Counterfactual Synthetic Control (CSC) models' utility in analyzing impactful policies. Our novel

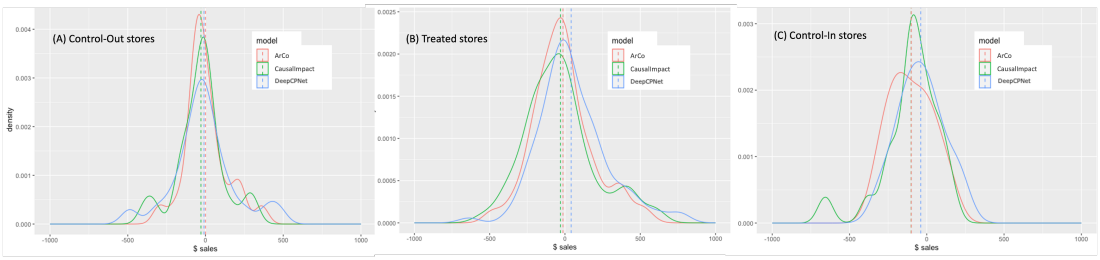


FIGURE 2.—Density distribution of the estimated ATE by the group of stores and per model for the first week of treatment. The dashed line represents the mean ATT/ATE_{ctr} for all series of the group.

GFM-DNN-based engine for counterfactual prediction overcomes this by using a synthetic control approach to learn from nonlinearly related and unrelated time series at different scales. It effectively discerns complex patterns and accurately estimates the ATE even with spillover effects where the SUTVA isn't fully met.

Recent research employing neural network forecasting for causal effect estimation has not incorporated GFM in policy evaluation contexts, where post-intervention data may distort ATE estimates. To the best of our knowledge, our study represents the inaugural effort to address this literature gap, theoretically and empirically. Our approach facilitates a more sophisticated analysis of intervention effects, better mirroring real-world conditions by easing some rigid assumptions typical of traditional methods. This innovation marks a significant advancement in our field. In our empirical analysis, we explore the effects of a prevalent marketing tactic—substantially discounting a product, in this case, milk, as part of a loss leader strategy. Our findings suggest that if our methodology is absent, one might erroneously conclude that this approach does not effectively boost sales.

In this paper, we have expanded on recent formal advancements regarding the consistency of local ReLU or smooth feedforward network estimators (Farrell et al., 2021, Fabozzi et al., 2019, Fallahgoul et al., 2024), extending these findings to Global Forecasting Methods for Feedforward Neural Network (GFM-FFNN) estimators. Our work significantly contributes to the burgeoning integration of global deep learning methodologies within theoretical and practical research, particularly in empirical causality investigations in social sciences, including economics.

The findings of this study are currently confined to scenarios involving single interventions. We plan to extend this to a staggered adaption design. Another promising avenue for

further investigation is extending the consistency proof to neural network classes beyond Feedforward Neural Networks (FFNN), such as Long Short-Term Memory (LSTM) networks. Given their demonstrated efficacy in time series forecasting, particularly in capturing time-variation features, LSTMs may offer more accurate forecasting of counterfactuals, thus broadening the scope.

REFERENCES

ABADI ET AL., MARTÍN (2015): "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," Software available from tensorflow.org. [20]

ABADIE, ALBERTO (2021): "Using synthetic controls: Feasibility, data requirements, and methodological aspects," *Journal of Economic Literature*, 59 (2), 391–425. [11]

ABADIE, ALBERTO, ALEXIS DIAMOND, AND JENS HAINMUELLER (2010): "Synthetic control methods for comparative case studies: Estimating the effect of California's tobacco control program," *Journal of the American statistical Association*, 105 (490), 493–505. [2, 11]

ANTHONY, MARTIN AND PETER L BARTLETT (2009): "Neural Network Learning: Theoretical Foundations," . [31]

ANTONELLI, JOSEPH AND BRENDEN BECK (2020): "Heterogeneous causal effects of neighborhood policing in New York City with staggered adoption of the policy," *arXiv preprint arXiv:2006.07681*. [9]

ARKHANGELSKY, DMITRY, SUSAN ATHEY, DAVID A. HIRSHBERG, GUIDO W. IMBENS, AND STEFAN WAGER (2021): "Synthetic Difference-in-Differences," *American Economic Review*, 111 (12), 4088–4118. [2]

ATHEY, SUSAN, MOHSEN BAYATI, NIKOLAY DOUDCHENKO, GUIDO IMBENS, AND KHASHAYAR KHOSRAVI (2021): "Matrix completion methods for causal panel data models," *Journal of the American Statistical Association*, 1–15. [2]

ATHEY, SUSAN AND GUIDO W IMBENS (2006): "Identification and inference in nonlinear difference-in-differences models," *Econometrica*, 74 (2), 431–497. [2]

BAJARI, PATRICK, BRIAN BURDICK, GUIDO W IMBENS, LORENZO MASOERO, JAMES MCQUEEN, THOMAS RICHARDSON, AND IDO M ROSEN (2021): "Multiple randomization designs," *arXiv preprint arXiv:2112.13495*. [3]

BANDARA, KASUN, CHRISTOPH BERGMEIR, SAM CAMPBELL, DEBORAH SCOTT, AND DAN LUBMAN (2020a): "Towards Accurate Predictions and Causal 'What-if' Analyses for Planning and Policy-making: A Case Study in Emergency Medical Services Demand," in *IJCNN*, IEEE, 1–10. [20]

BANDARA, KASUN, CHRISTOPH BERGMEIR, AND HANSIKA HEWAMALAGE (2020b): "LSTM-MSNet: Leveraging Forecasts on Sets of Related Time Series with Multiple Seasonal Patterns," *IEEE TNNLS*. [27]

BANDARA, KASUN, CHRISTOPH BERGMEIR, AND SLAWEK SMYL (2020c): "Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach," *Expert Systems with Applications*, 140, 112896. [3, 17, 18, 27]

BANDARA, KASUN, PEIBEI SHI, CHRISTOPH BERGMEIR, HANSIKA HEWAMALAGE, QUOC TRAN, AND BRIAN SEAMAN (2019): "Sales demand forecast in e-commerce using a long short-term memory neural network methodology," in *ICONIP*, Springer, 462–474. [27]

BRODERSEN, KAY H, FABIAN GALLUSSER, JIM KOEHLER, NICOLAS REMY, AND STEVEN L SCOTT (2015): "Inferring causal impact using Bayesian structural time-series models," *The Annals of Applied Statistics*, 9 (1), 247–274. [2, 12]

CARVALHO, CARLOS, RICARDO MASINI, AND MARCELO C MEDEIROS (2018): "Arco: an artificial counterfactual approach for high-dimensional panel time-series data," *Journal of Econometrics*, 207 (2), 352–380. [2, 12, 22, 34]

CERQUA, AUGUSTO, MARCO LETTA, AND FIAMMETTA MENCHETTI (2022): "Losing Control (Group)? The Machine Learning Control Method for Counterfactual Forecasting," *The Machine Learning Control Method for Counterfactual Forecasting (December 30, 2022)*. [3, 6, 9]

CHERNOZHUKOV, VICTOR, KASPAR WÜTHRICH, AND YINCHU ZHU (2021): "An exact and robust conformal inference method for counterfactual and synthetic controls," *Journal of the American Statistical Association*, 1–16. [2, 6, 10, 20, 21, 38]

CLEVELAND, R.B., W.S. CLEVELAND, J.E. MCRAE, AND I. TERPENNING (1990): "STL: A Seasonal-Trend Decomposition Procedure Based on Loess," *J. Off. Stat.*, 6 (1), 3–33. [19]

CONLEY, TIMOTHY G AND CHRISTOPHER R TABER (2011): "Inference with "difference in differences" with a small number of policy changes," *The Review of Economics and Statistics*, 93 (1), 113–125. [2]

DOUDCHENKO, NIKOLAY AND GUIDO W IMBENS (2016): "Balancing, regression, difference-in-differences and synthetic control methods: A synthesis," Tech. rep., National Bureau of Economic Research. [2, 11]

ELMAN, JEFFREY L (1990): "Finding structure in time," *Cognitive science*, 14 (2), 179–211. [17]

FABOZZI, FRANK J, HASAN FALLAHGOUL, VINCENTIUS FRANSTIANTO, AND GREGOIRE LOEPER (2019): "Towards Explaining Deep Learning: Asymptotic Properties of ReLU FFN Sieve Estimators," *Available at SSRN 3499324*. [5, 27, 41]

FALLAHGOUL, HASAN, VINCENTIUS FRANSTIANTO, AND XIN LIN (2024): "Asset pricing with neural networks: Significance tests," *Journal of Econometrics*, 238 (1), 105574. [5, 21, 22, 27, 28, 29, 31, 32, 34, 41]

FARRELL, MAX H, TENGYUAN LIANG, AND SANJOG MISRA (2021): "Deep neural networks for estimation and inference," *Econometrica*, 89 (1), 181–213. [4, 5, 27, 30, 31, 32, 41]

FERMAN, BRUNO AND CRISTINE PINTO (2016): "Revisiting the synthetic control estimator," *Munich Personal RePEc Archive - MPRA*. [12]

GOBILLON, LAURENT AND THIERRY MAGNAC (2016): "Regional policy evaluation: Interactive fixed effects and synthetic controls," *Review of Economics and Statistics*, 98 (3), 535–551. [2, 11]

GRECOV, PRISCILA, ANKITHA NANDIPURA PRASANNA, KLAUS ACKERMANN, SAM CAMPBELL, DEBBIE SCOTT, DAN I LUBMAN, AND CHRISTOPH BERGMEIR (2022): "Probabilistic causal effect estimation with global neural network forecasting models," *IEEE Transactions on Neural Networks and Learning Systems*. [34, 37]

HARTFORD, JASON, GREG LEWIS, KEVIN LEYTON-BROWN, AND MATT TADDY (2017): "Deep IV: A flexible approach for counterfactual prediction," in *ICML*, 1414–1423. [4]

HEWAMALAGE, HANSIKA, CHRISTOPH BERGMEIR, AND KASUN BANDARA (2021a): "Global models for time series forecasting: A simulation study," *Pattern Recognition*, 108441. [27]

——— (2021b): "Recurrent neural networks for time series forecasting: Current status and future directions," *International Journal of Forecasting*, 37 (1), 388–427. [17, 18, 20]

HOCHREITER, SEPP AND JÜRGEN SCHMIDHUBER (1997): "Long short-term memory," *Neural computation*, 9 (8), 1735–1780. [17]

HOEFFDING, WASSILY (1994): "Probability inequalities for sums of bounded random variables," in *The collected works of Wassily Hoeffding*, Springer, 409–426. [25]

HOREL, ENGUERRAND AND KAY GIESECKE (2020): "Significance tests for neural networks," *The Journal of Machine Learning Research*, 21 (1), 9291–9319. [27]

HUTTER, FRANK, HOLGER H HOOS, AND KEVIN LEYTON-BROWN (2011): "Sequential model-based optimization for general algorithm configuration," in *International conference on learning and intelligent optimization*, Springer, 507–523. [20]

JANUSCHOWSKI, TIM, JAN GASTHAUS, YUYANG WANG, DAVID SALINAS, VALENTIN FLUNKERT, MICHAEL BOHLKE-SCHNEIDER, AND LAURENT CALLOT (2020): "Criteria for classifying forecasting methods," *Int. J of Forecasting*, 36 (1), 167–177. [3]

JOHANSSON, FREDRIK, URI SHALIT, AND DAVID SONTAG (2016): "Learning representations for counterfactual inference," in *International conference on machine learning*, 3020–3029. [4]

JOHARI, RAMESH, HANNAH LI, INESSA LISKOVICH, AND GABRIEL Y WEINTRAUB (2022): "Experimental design in two-sided platforms: An analysis of bias," *Management Science*. [5]

KINGMA, DIEDERIK P AND JIMMY BA (2014): "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*. [17, 20]

KLINENBERG, DANNY (2021): "Synthetic Control with Time Varying Coefficients: A State Space Approach with Bayesian Shrinkage," *Available at SSRN 3774589*. [2]

LIM, BRYAN (2018): "Forecasting treatment responses over time using recurrent marginal structural networks," in *NeurIPS*, 7483–7493. [4]

LIU, MIN, JIALIANG MAO, AND KANG KANG (2021): "Trustworthy and Powerful Online Marketplace Experimentation with Budget-split Design," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 3319–3329. [5]

MAKRIDAKIS, SPYROS, EVANGELOS SPILIOTIS, AND VASSILIOS ASSIMAKOPOULOS (2020): "The M4 Competition: 100,000 time series and 61 forecasting methods," *International Journal of Forecasting*, 36 (1), 54–74. [27]

MENCHETTI, FIAMMETTA, FABRIZIO CIPOLLINI, AND FABRIZIA MEALLI (2023): "Combining counterfactual outcomes and ARIMA models for policy evaluation," *The Econometrics Journal*, 26 (1), 1–24. [3, 6, 9]

MOJIR, NAVID AND K SUDHIR (2021): "A model of multipass search: Price search across stores and time," *Management Science*, 67 (4), 2126–2150. [36]

MONTERO-MANSO, PABLO AND ROB J HYNDMAN (2021): "Principles and algorithms for forecasting groups of time series: Locality and globality," *International Journal of Forecasting*, 37 (4), 1632–1653. [3, 5, 22, 23, 24, 25, 27]

NEYMAN, JERZY S (1923): "On the application of probability theory to agricultural experiments. essay on principles. section 9.(tlanslated and edited by dm dabrowska and tp speed, statistical science (1990), 5, 465-480)," *Annals of Agricultural Sciences*, 10, 1–51. [6]

NICKL, RICHARD AND BENEDIKT M PÖTSCHER (2007): "Bracketing metric entropy rates and empirical central limit theorems for function classes of Besov-and Sobolev-type," *Journal of Theoretical Probability*, 20, 177–199. [28]

PAPADOGEORGOU, GEORGIA, FIAMMETTA MENCHETTI, CHRISTINE CHOIRAT, JASON H WASFY, CORWIN M ZIGLER, AND FABRIZIA MEALLI (2023): "Evaluating federal policies using Bayesian time series models: estimating the causal impact of the hospital readmissions reduction program," *Health Services and Outcomes Research Methodology*, 1–19. [9]

PENG, ANBANG, XIAOLI ZHANG, WEI XU, AND YUANYANG TIAN (2022): "Effects of training data on the learning performance of LSTM network for runoff simulation," *Water Resources Management*, 36 (7), 2381–2394. [34]

PITMAN, JIM (2012): *Probability*, Springer Science & Business Media. [25]

POGGIO, TOMASO, HRUSHIKESH MHASKAR, LORENZO ROSASCO, BRANDO MIRANDA, AND QIANLI LIAO (2017): "Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review," *International Journal of Automation and Computing*, 14 (5), 503–519. [28, 29, 30, 32]

POULOS, JASON (2017): "RNN-based counterfactual timeseries prediction," *arXiv preprint arXiv:1712.03553*. [2, 4]

RUBIN, DONALD B (1974): "Estimating causal effects of treatments in randomized and nonrandomized studies," *Journal of educational Psychology*, 66 (5), 688. [2, 6]

——— (1980): "Randomization analysis of experimental data: The Fisher randomization test comment," *Journal of the American statistical association*, 75 (371), 591–593. [9]

SALINAS, DAVID, VALENTIN FLUNKERT, JAN GASTHAUS, AND TIM JANUSCHOWSKI (2020): "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *Int. J of Forecasting*, 36 (3), 1181–1191. [3]

SHI, CLAUDIA, DAVID BLEI, AND VICTOR VEITCH (2019): "Adapting neural networks for the estimation of treatment effects," in *NeurIPS*, 2507–2517. [4]

SMYL, SLAWEK (2020): "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting," *International Journal of Forecasting*, 36 (1), 75–85. [3]

SPILIOTIS, E, V ASSIMAKOPOULOS, S MAKRIDAKIS, AND V ASSIMAKOPOULOS (2022): "The M5 Accuracy competition: Results, findings and conclusions," *Int. J. Forecast.* [27]

YAROTSKY, DMITRY (2017): "Error bounds for approximations with deep ReLU networks," *Neural Networks*, 94, 103–114. [32]