Time-Aware Synthetic Control

Anonymous Author(s)
Affiliation
Address
email

Abstract

The synthetic control (SC) framework is a key tool for observational causal inference in time-series panel data, common in healthcare and clinical research. Although the data analyzed by SC are inherently time-seires, most SC approaches are invariant to permutations of the time indices in the pre-intervention data. In this work, we suggest Time-Aware Synthetic Control (TASC), which models the observations using a linear state space model and performs counterfactual inference using the Kalman filter and RTS smoothing. TASC ensures that the data maintains a low-rank signal with latent factors evolving gradually over time. As an initial demonstration, we apply the TASC approach to a case study on California's healthcare policy (Proposition 99). Our method showed promising results in placebo tests, indicating its potential applicability in a broader range of healthcare contexts.

1 Introduction

Synthetic Control (SC) is a popular method in observational causal inference. Often called as a natural extension of the Difference-in-Differences method (D-in-D, [1]), SC aims to evaluate the effects of an intervention more accurately by creating synthetic counterfactual data. The first application was measuring the economic impact of the 1960's terrorist conflict in Basque Country, Spain (a *target unit*) by combining GDP data from other Spanish regions (*donor units*) prior to the conflict to construct a *synthetic* GDP data for Basque Country in the counterfactual world without the conflict [2]. Rather than selecting the nearest neighbor as in D-in-D, the syntheitc control estimate is computed as a linear combination of donor units. SC is becoming more and more popular with expanding range of applications, including economics [2, 3, 4], political sciences [5, 6], social sciences [7, 8], healthcare [9, 8, 10], to name a few.

A fundamental assumption of SC is that **the time-series panel data arises from a latent variable model, producing an (approximately) low-rank matrix**. Various latent variable models have been adopted, including the linear factor model [2, 3, 5], latent variable models with Lipschitz-continuous functions [11, 12], and the interactive fixed-effect model [13]. Despite differences in these models, they commonly assume that the observation matrix can be well-approximated by a low-rank matrix, implying that the latent factors are simpler than the observed data. These methods typically leverage the matrix's spectrum analysis: RSC [11] uses principal component regression, while [14] frames SC as a nuclear-norm minimization problem. However, due to their reliance on spectrum analysis, these approaches are time-agnostic, meaning permutations of pre-intervention columns do not impact the results.

Remind that **the panel data analyzed using SC is inherently time-series data**, and we can use this in learning the SC model. Some approaches were proposed to incorporate the ordered nature of time indices using a state-space model. [15] designed the state vector to include elements like SC weights, local linear trends, and seasonality, and the observation is the target unit's outcomes. [16, 17] further simplified this structure and only take SC weights as a latent state. These approaches model the observation as a time series of the target unit, but does not explicitly connect their results

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

to the low-rank structure of the panel data (SC weights are n dimensional, where n is the number of donors).

Our contribution lies in embedding SC panel data within a state-space model to simultaneously harness both the low-rank and time-series properties of the data. In Section 3, we introduce our model and outline a simple MAP approach to learn the model, employing Kalman filtering and Rauch–Tung–Striebel smoothing. In Section 4, we apply our method to a classic health policy evaluation case (Proposition 99) demonstrating enhanced performance and interpretability relative to traditional methods. This approach offers a principled framework for integrating causal inference with time-series analysis, which is particularly valuable in healthcare and clinical research settings.

2 Background

2.1 Synthetic Control Methods

The time-series panel dataset for SC consists of the following components. Let $Y_{i,t} \in \mathbb{R}$ be the observation from i-th unit at time t. We have n (untreated) donor units indexed by $i \in \{1, \ldots, n\}$ and a (treated) target unit with index 0. The untreated observation matrix is of size $(n+1) \times T$, where the target unit's values after $t > T_0$ is missing due to the treatment happening at time T_0. Figure 1 illustrates the general structure of an SC dataset, where the superscript $-$ denotes the pre-intervention period and the superscript $+$ denotes the post-intervention period.

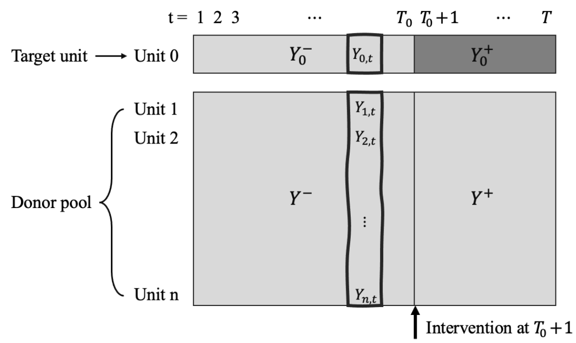

Figure 1: General data structure for synthetic control. The donor pool ($Y = [y_1, \ldots, y_n]^\top$) and the target unit ($x_0$) are divided into pre- and post-intervention periods.

Based on this data structure, we define SC family of methods (Algorithm 1) as follows. SC first learns the relationship between the target unit and donors using the pre-intervention data. For example, this ($\mathcal{M}$) can be a *vertical* regression where the pre-intervention column vectors $Y_{1:n,t}$ become input features for the label $Y_{0,t}$, for all $t \in [T_0]$. Then, SC uses this knowledge (f) to project the post-intervention donor data Y^+ and predict $\hat{Y}_0^+$. Finally, the causal effect of the intervention for the target unit is estimated as $Y_0^+ - \hat{Y}_0^+$.

Algorithm 1 Synthetic Control Family of Methods

Data: Target unit's pre-intervention data $Y_0^- \in \mathbb{R}^{T_0}$, Donor data $Y = [Y^-, Y^+] \in \mathbb{R}^{n \times T}$
Result: Counterfactual prediction $\hat{Y}_0^+$, SC weights f
1. Learn $f = \mathcal{M}(Y_0^-, Y^-, Y^+)$ /* the use of Y^+ is optional */
2. Project $\hat{Y}_0^+ = f(Y^+)$
3. Infer the estimated causal effect of the intervention for the target is $Y_0^+ - \hat{Y}_0^+$

The core learning algorithm for SC is the $\mathcal{M}$ part. The first SC algorithm suggested by [2] assumes a linear factor model

$$Y_{i,t} = \delta_t + \theta_t Z_i + \lambda_t \mu_i + \epsilon_{i,t}, \tag{1}$$

where δ_t is a time trend, $Z_i \in \mathbb{R}^p$ and $\mu_i \in \mathbb{R}^q$ are vector of observed and unobserved predictors, with coefficients θ_t and μ_i, and $\epsilon_{i,t}$ is the noise. Then, it uses the following optimization approach: $\min_{f \in \mathbb{R}^n} ||Y_0^- - Y^{-\top} f||^2 \quad s.t. \sum_i f_i = 1, 0 \leq f_i \leq 1 \quad \forall i \in [n]$. This can be easily extended to have regularizers (such as Lasso $||f||_1$ or Ridge $||f||_2^2$ added to the minimization objective) and/or dropping the simplex constraint. Robust Synthetic Control (RSC, [11]) uses PCA to keep only the top few singular values in donor data prior to the optimization step (essentially performing Principal Component Regression).

2.2 Approximately Low-Rank Observation with Temporal Structure

The classical SC method is based on a linear factor model, as described in Equation (1). This model implies that the signal component of the matrix has a rank no more than $p+q+1$. When this quantity is considerably smaller than the matrix's full rank, the observation matrix becomes approximately low rank. This has inspired a range of SC algorithms. Several SC algorithms [2, 3, 18, 19] employ simplex constraints or regularizers to minimize the number of active donor units; RSC [11, 12] uses principal component regression; and [14] frames SC as a problem of nuclear norm minimization.

Another characteristic of the panel data used in SC is its time-series nature. Despite this, many SC algorithm variants, including all those mentioned in the previous paragraph, remain invariant to permutations of time indices in pre-intervention data. To address this limitation, some researchers have introduced algorithms that understand temporal structure by utilizing state space models [15, 16, 17]. The central concept of these approaches is to allow SC weights to change over time, defining the target unit's time series as an observation (scalar) and the SC weights (potentially alongside additional components) as latent states (at least n-dimensional vector). These modeling approaches do not necessarily ensure that the observation matrix is approximately low-rank.

3 State-Space Model for Time-Aware Synthetic Control

We assume the following state space model:

$$x_t = Ax_{t-1} + q_{t-1} \quad q_{t-1} \sim \mathcal{N}(0, Q), \tag{2}$$

$$y_t = Hx_t + r_t \quad, r_t \sim \mathcal{N}(0, R). \tag{3}$$

where we assume the initial hidden state x_0 to be drawn from $\mathcal{N}(m_0, P_0)$ (i.e., $x_0 \sim \mathcal{N}(m_0, P_0)$.)

The model parameters are $\theta = \{A, H, Q, R, m_0, P_0\}$, where $A \in \mathbb{R}^{d\times d}$, $H \in \mathbb{R}^{(n+1)\times d}$, $Q \in \mathbb{R}^{d\times d}$, $R \in \mathbb{R}^{(n+1)\times(n+1)}$, $m_0 \in \mathbb{R}^d$, $P_0 \in \mathbb{R}^{d\times d}$, and $d \ll min(n, T)$. This is a classical linear Gaussian model, and we set all covariance matrices Q, R, and P_k be positive semi-definite. If desired, we may constrain that the noise covariance matrices Q and R are diagonal with non-zero diagonal entries to reduce the number of parameters. The advantage of this model is that the output $Y = [y_1, \ldots, y_T]$ will be an approximately low-rank matrix (with the approximate rank being d), and the learning algorithm will no longer be agnostic to the permutations of pre-intervention time indices.

Based on this model, **we propose two versions of TASC Algorithms to obtain MAP estimators: $TASC_{pre}$ and $TASC_{full}$.** Both are Expectation-Maximization(EM) algorithms to learn the parameters: we start by randomly choosing the parameters θ, and then keep updating as we get the data. $TASC_{pre}$ only uses the pre-intervention data to learn the parameters, and then use Kalman filter and RTS smoother to predict the counterfactual outcomes. Note this prediction phase still utilizes post-intervention data, but they are not used for updating parameters. $TASC_{full}$ continues to update the parameters using post-intervention observations as well. To do so, we augment the target's post-intervention observations with the prediction based on the most recently updated parameter. A full description of algorithms are deferred to Appendix B.

3.1 Comparison to RSC

Assume that matrix entries follow a latent variable model $Y_{i,t} = g(\theta_i, \rho_t) + \epsilon_{i,t}$ where g is a dot product, θ_i and ρ_t are d dimensional latent vectors characterizing i-th unit and t-th time, and $\epsilon_{i,t}$ is observation noise. In RSC [11], θ_i and ρ_t are derived from PCA: $X = \sum_{i=1}^{min(n,T)} s_i u_i v_i^\top = \sum_{i=1}^{d} s_i u_i v_i^\top + \sum_{i=d+1}^{min(n,T)} s_i u_i v_i^\top$, where s_i is singular values in decreasing order. By defining $\tilde{U}$ to have $s_i^{1/2} u_i$ for $i \le d$ as columns and $\tilde{V}^\top$ to have $s_i^{1/2} v_i$ for $i \le d$ as rows, the rows of $\tilde{U}$ can be interpreted as θ_i and the columns of $\tilde{V}$ as ρ_t. Similary, the $TASC$ model suggests a decomposition $Y = HX + E$, where the columns of X are hidden states x_t and the columns of E are observation noise r_t. Here, the rows of H are analogous to θ_i and the columns of X are to ρ_t. Both $\tilde{U}\tilde{V}^\top$ and HX are exactly low-rank matrices, but with differ in how we separate the noise. The difference comes from the learning objectives: RSC's approach minimizes the size of the noise matrix (in terms of spectral norm), whereas $TASC$ focuses on making sure the time-features ρ_t (or x_t in our model) evolves gradually over time with a trend (A). As a result, the noise portion in RSC becomes rank $min(n, T) - d$, whereas E is almost surely full rank (omnidirectional).

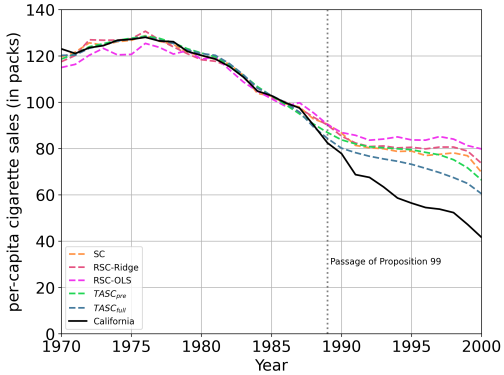

Figure 2: Per-capita cigarette sales in packs in California (black solid line), and estimated counterfactual values from SC (dotted lines).

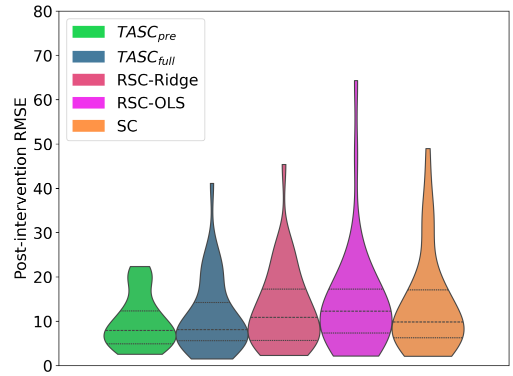

Figure 3: Post-intervention RMSE from placebo test for different SC algorithms

4 Evaluating Effect of Proposition 99 in California

We illustrate our method on the Proposition 99 case [3], which raised California's cigarette tax in 1988 and was followed by a drop in sales (black line, Figure 2). To test if this decline was policy-driven, we compare counterfactuals from SC [3], RSC [11], and our $TASC_{pre}$ and $TASC_{full}$ models[1]. All predictions lie above the observed trend, with the gap capturing the policy's effect.

The next question is which SC estimates are most reliable. Since counterfactuals are unobservable, we use a *placebo test*: predicting each donor's time series from the others. A valid SC method should also accurately reproduce untreated outcomes. Figure 3 shows post-intervention root mean squared error (RMSE) from the placebo test across different implementations of SC. Among these, $TASC_{pre}$ achieves the lowest RMSE, followed by $TASC_{post}$, RSC-Ridge, SC, and RSC-OLS. This ordering suggests that the $TASC$ estimates in Figure 2 likely yield counterfactual predictions with smaller error and may therefore be closer to the true counterfactual.

In Appendix D, we present additional analyses, including a comparison highlighting the advantages of our *interpretable* latent factor model, practical strategies for improving the EM algorithm, and a dropout analysis to assess the robustness of California's counterfactual prediction.

5 Conclusion and Future Work

In this work, we introduced a state-space framework to model synthetic control type of data, and a family of algorithms (TASC) to learn the model. With this approach, we enforce synthetic control algorithm to be aware of the temporal structure of data, which was ignored in many variations of synthetic control algorithms. We demonstrated $TASC$ on the classical Proposition 99 case study, and the placebo test showed promising result that $TASC$ may be a favorable choice in certain datasets over SC or RSC. We suspect that a pronounced temporal trend is the key for $TASC$ to succeed, which is a common case for a lot of health and clinical time-series panel data.

Looking forward, we see several directions for extension. In terms of the model and algorithms, we can incorporate multiple auxiliary time series, improve the parameter learning algorithm, and also consider non-linear state-space models. From a theoretical perspective, it would be valuable to identify conditions under which $TASC$ is preferable to other SC variants, and to determine if these conditions are testable. For the applications, we can apply TASC to clinical trial datasets or electronic health records. With the promising initial result, we envision that $TASC$ could serve as a robust framework for integrating causal inference with time-series modeling.

[1]SC uses the relative importance matrix $V = I$ (See [3] for more information, it is equivalent to linear regression with simplex constraint introduced in Section 2.1); RSC uses top $d = 2$ singular values in noise filtering step and Ridge regression coefficient was set to 0.1 (See [11] for more information); $TASC$ uses $d = 2$ for hidden state dimension

References

[1] David Card and Alan B Krueger. Minimum wages and employment: A case study of the fast food industry in New Jersey and Pennsylvania. *American Economic Review*, 90(5):1397–1420, 2000.

[2] Alberto Abadie and Javier Gardeazabal. The economic costs of conflict: A case study of the Basque Country. *American Economic Review*, 93(1):113–132, 2003.

[3] Alberto Abadie, Alexis Diamond, and Jens Hainmueller. 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, 2010.

[4] Alberto Abadie and Jérémy L'Hour. A penalized synthetic control estimator for disaggregated data. *Journal of the American Statistical Association*, 116(536):1817–1834, 2021.

[5] Alberto Abadie, Alexis Diamond, and Jens Hainmueller. Comparative politics and the synthetic control method. *American Journal of Political Science*, 59(2):495–510, 2015.

[6] Noémi Kreif, Richard Grieve, Dominik Hangartner, Alex James Turner, Silviya Nikolova, and Matt Sutton. Examination of the synthetic control method for evaluating health policies with multiple treated units. *Health economics*, 25(12):1514–1528, 2016.

[7] Michael W Robbins, Jessica Saunders, and Beau Kilmer. A framework for synthetic control methods with high-dimensional, micro-level data: evaluating a neighborhood-specific crime intervention. *Journal of the American Statistical Association*, 112(517):109–126, 2017.

[8] Giacomo Vagni and Richard Breen. Earnings and income penalties for motherhood: estimates for British women using the individual synthetic control method. *European Sociological Review*, 37(5):834–848, 2021.

[9] Kristian Thorlund, Louis Dron, Jay JH Park, and Edward J Mills. Synthetic and external controls in clinical trials–a primer for researchers. *Clinical epidemiology*, pages 457–467, 2020.

[10] Dennis Shen, Anish Agarwal, Vishal Misra, Bjoern Schelter, Devavrat Shah, Helen Shiells, and Claude Wischik. Obtaining personalized predictions from a randomized controlled trial on alzheimer's disease. *Scientific Reports*, 15(1):1671, 2025.

[11] Muhammad Amjad, Devavrat Shah, and Dennis Shen. Robust synthetic control. *Journal of Machine Learning Research*, 19(22):1–51, 2018.

[12] Muhammad Amjad, Vishal Misra, Devavrat Shah, and Dennis Shen. mRSC: Multi-dimensional robust synthetic control. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(2):1–27, 2019.

[13] Yiqing Xu. Generalized synthetic control method: Causal inference with interactive fixed effects models. *Political Analysis*, 25(1):57–76, 2017.

[14] Susan Athey, Mohsen Bayati, Nikolay Doudchenko, Guido Imbens, and Khashayar Khosravi. Matrix completion methods for causal panel data models. *Journal of the American Statistical Association*, 116:1716–1730, 2021.

[15] Kay H Brodersen, Fabian Gallusser, Jim Koehler, Nicolas Remy, and Steven L Scott. Inferring causal impact using bayesian structural time-series models. *The Annals of Applied Statistics*, 9(1):247–274, 2015.

[16] Danny Klinenberg. Synthetic control with time varying coefficients a state space approach with bayesian shrinkage. *Journal of Business & Economic Statistics*, 41(4):1065–1076, 2023.

[17] Junzhe Shao, Mingzhang Yin, Xiaoxuan Cai, and Linda Valeri. Generalized synthetic control method with state-space model. In *NeurIPS 2022 Workshop on Causality for Real-world Impact*, 2022.

[18] Nikolay Doudchenko and Guido W Imbens. Balancing, regression, difference-in-differences and synthetic control methods: A synthesis, 2016. NBER Working Paper 22791.

[19] Victor Chernozhukov, Kaspar Wüthrich, and Yinchu Zhu. An exact and robust conformal inference method for counterfactual and synthetic controls. *Journal of the American Statistical Association*, 116(536):1849–1864, 2021.

A Basic Algorithms

In this section, we provide the full pseudocode for the basic algorithms comprising the EM approach: Kalman Filter (Algorithm 2), RTS Smoother (Algorithm 3), and the M-step (Algorithm 4).

Algorithm 2 Kalman Filter

Input : $y_k \in \mathbb{R}^{n+1}$, previous estimate m_{k-1}, P_{k-1}, current parameter $\theta = \{A, H, Q, R, m_0, P_0\}$
Output : m_k, P_k

```
m_{k|k-1} ← A m_{k-1}              /* prediction from the previous timestep k − 1 */
P_{k|k-1} ← A P_{k-1} A^⊤ + Q      /* prediction from the previous timestep k − 1 */
v_k ← y_k − H m_{k|k-1}
S_k ← H P_{k|k-1} H^⊤ + R
K_k ← P_{k|k-1} H^⊤ S_k^{-1}                                   /* Kalman Gain */
m_k ← m_{k|k-1} + K_k v_k                          /* Update after observing y_k */
P_k ← P_{k|k-1} − K_k S_k K_k^⊤                    /* Update after observing y_k */
```

Algorithm 3 RTS Smoother

Input : Kalman filter estimate m_k, P_k, smooted estimate m^s_{k+1}, P^s_{k+1}, current parameter $\theta = \{A, H, Q, R, m_0, P_0\}$
Output : m^s_k, P^s_k

```
m_{k+1|k} ← A m_k                 /* prediction from Kalman filter estimate m_k */
P_{k+1|k} ← A P_k A^⊤ + Q         /* prediction from Kalman filter estimate P_k */
G_k ← P_k A^⊤ P_{k+1|k}^{-1}
m^s_k ← m_k + G_k[m^s_{k+1} − m_{k+1|k}]   /* m^s_t = m_t for the last timestep t = T_0 or T */
P^s_k ← P_k + G_k[P^s_{k+1} − P_{k+1|k}]G_k^⊤   /* P^s_t = P_t for the last timestep t = T_0 or T */
return m^s_k, P^s_k, G_k
```

Algorithm 4 Parameter Update (M-Step)

Input : current parameter $\theta = \{A, H, Q, R, m_0, P_0\}$, length of the sequence T, RTS parameters m^s_k, P^s_k, G_k for all $k \in \{0, \dots, T\}$, observations y_k for all $k \in \{1, \dots, T\}$
Output : θ'

Define

$\Sigma = \frac{1}{T}\sum_{k=1}^{T} P^s_k + m^s_k {m^s_k}^\top$

$\Phi = \frac{1}{T}\sum_{k=1}^{T} P^s_{k-1} + m^s_{k-1} {m^s_{k-1}}^\top$

$B = \frac{1}{T}\sum_{k=1}^{T} y_k {m^s_k}^\top$

$C = \frac{1}{T}\sum_{k=1}^{T} P^s_k G_{k-1}^\top + m^s_k {m^s_{k-1}}^\top$

$D = \frac{1}{T}\sum_{k=1}^{T} y_k {y_k}^\top$

Update

$A' \leftarrow C\Phi^{-1}$

$H' \leftarrow B\Sigma^{-1}$

$Q' \leftarrow \mathrm{Diag}(\Sigma - 2CA^\top + A\Phi A^\top)$ `/* Diag(·) keeps only the diagonal elements of the input */`

$R' \leftarrow \mathrm{Diag}(D - 2BH^\top + H\Sigma H^\top)$

$m'_0 \leftarrow m^s_0$

$P'_0 \leftarrow P^s_0 + (m^s_0 - m_0)(m^s_0 - m_0)^\top$

return $\theta' = \{A', H', Q', R', m'_0, P'_0\}$

B Full description of Time-Award Synthetic Control (TASC) Algorithms

In this section, we disclose the derivation of the three versions of Kalman Synthetic Control:$TASC_{pre}$ (only using pre-intervention data for EM updates) and $TASC_{full}$ (using both pre- and post-intervention data for EM updates).

B.1 Pre-intervention Fit

For pre-intevention data, we can take the classical EM approach for a linear gaussian state-space model. The E-step comprises of a forwad pass (Kalman filter) and a backward pass (RTS smoothing). This gives us estimates m_k^s and P_k^s to define a lower bound for the posterior probability distribution.

Algorithm 5 $EM_{pre}(Y_{pre}; N)$, EM for Pre-intervention Fit

Data: Y_{pre} where (i,j)-th element is $y_{i,t}$ $\forall(i,t) \in [0:n] \times [1:T_0]$ (pre-intervention data from the target and donors)
Result: $\theta = \{A, H, Q, R, m_0, P_0\}$
Initialize $\theta^{(0)}$
for $i \leftarrow 1$ **to** N **do**
  for $k \leftarrow 1$ **to** T_0 **do**
    Update m_k, P_k via Kalman filtering with $\theta^{(i-1)}$; `/* forward pass */`
  end
  for $k \leftarrow T_0 - 1$ **to** 0 **do**
    Update m_k^s, P_k^s, G_k via RTS Smoothing with $\theta^{(i-1)}$; `/* backward pass */`
  end
  Update $\theta^{(i)}$ via the M-step of EM
end
return $\theta^{(N)}$

$TASC_{pre}$ utilizes Algorithm 5 to learn the parameters θ from the pre-intervention data. With this fixed parameters, $TASC_{pre}$ uses Kalman filter and RTS smoother to estimate the internal states m_k^s, P_k^s, and then translate these to finally estimate the post-intervention time series. However, this is impossible without a special treatment since the first element of y_k (which belongs to the target unit) is missing. To handle this, we deem that the target unit's data is missing, and separate the donor portion of the data and parameters:

$$y_t = \begin{bmatrix} y_{t,1} \\ y_{t,2} \end{bmatrix}, r_t = \begin{bmatrix} r_{t,1} \\ r_{t,2} \end{bmatrix}, H = \begin{bmatrix} h_1^\top \\ H_2 \end{bmatrix}, \text{ and } R = \begin{bmatrix} r_1 & 0 \\ 0 & R_2 \end{bmatrix},$$

where $y_{t,2}, r_{t,2} \in \mathbb{R}^n$, $H_2 \in \mathbb{R}^{n \times d}$, and $R_2 \in \mathbb{R}^{n \times n}$.

Then, we redefine the observation model (i.e., Equation (3)) to be the following

$$y_{t,2} = H_2 x_t + r_{t,2},$$

where $r_{t,2} \sim \mathcal{N}(0, R_2)$. With this new model, the post-intervention observations will not inform the target-related parameters: h_1 and r_1. This is equivalent to setting $r_1 \to \infty$ in the original model.

Algorithm 6 shows the Kalman filtering process with infinite variance for the target observation (i.e., $r_1 \to \infty$). Note that

$$\begin{aligned} K_k &= P_{k|k-1} H^\top S_k^{-1} \\ &= \begin{bmatrix} 0 & P_{k|k-1}^{-1} H_2^\top (H_2 P_{k|k-1} H_2^\top + R_2)^{-1} \end{bmatrix}, \end{aligned}$$

and the update on m_k is

$$\begin{aligned} m_k &= m_{k|k-1} + K_k v_k \\ &= m_{k|k-1} + P_{k|k-1}^{-1} H_2^\top (H_2 P_{k|k-1} H_2^\top + R_2)^{-1} v_{k,2}, \end{aligned}$$

(Note: $v_{k,2} \in \mathbb{R}^n$ is the last n elements of v_k—i.e., $v_{k,2} = y_{k,2} - H_2 m_{k|k-1}$) , and the update on P_k is

$$
\begin{aligned}
P_k &= P_{k|k-1} + K_k S_k K_k^\top \\
&= P_{k|k-1} + P_{k|k-1} H_2^\top (H_2 P_{k|k-1} H_2^\top + R_2)^{-1} H_2 P_{k|k-1}^\top .
\end{aligned}
$$

Algorithm 6 Kalman Filter with Infinite Variance

Input : $y_k \in \mathbb{R}^{n+1}$ with the target(first) element missing, previous estimate m_{k-1}, P_{k-1}, current parameter $\theta' = \{A, H, Q, R, m_0, P_0\}$, where $R'_{1,1} = \infty$

Output : m_k, P_k

Define h_1, H_2, R_2 from $H = \begin{bmatrix} h_1^\top \\ H_2 \end{bmatrix}, R' = \begin{bmatrix} \infty & 0 \\ 0 & R_2 \end{bmatrix}$

$y_k \leftarrow [h_1^\top m_{k|k-1}, y_{1,k}, \ldots, y_{n,k}]^\top$ `/* augment target values */`

$m_{k|k-1} \leftarrow A m_{k-1}$ `/* prediction from the previous timestep` $k-1$ `*/`

$P_{k|k-1} \leftarrow A P_{k-1} A^\top + Q$ `/* prediction from the previous timestep` $k-1$ `*/`

$v_k \leftarrow y_k - H m_{k|k-1}$ `/* the first element is zero */`

$S_k \leftarrow H P_{k|k-1} H^\top + R'$

$S_k^{-1} \leftarrow \begin{bmatrix} 0 & 0 \\ 0 & (H_2 P_{k|k-1} H_2^\top + R_2)^{-1} \end{bmatrix}$ `/* by Schur Complement */`

$K_k \leftarrow P_{k|k-1} H^\top S_k^{-1}$

$m_k \leftarrow m_{k|k-1} + K_k v_k$ `/* Update after observing` y_k `*/`

$P_k \leftarrow P_{k|k-1} - K_k S_k K_k^\top$ `/* Update after observing` y_k `*/`

With the infinite variance, post-intervention target time series do not affect the outcome of Kalman filtering, hence we can set it to any value (such as zero for the first element in $y_k \leftarrow [0, y_{1,k}, \ldots, y_{n,k}]^\top$). The RTS Smoother algorithm remains the same, as it does not use R or y_k as an input—it only utilizes m_k, P_k estimates from the Kalman filter in addition to the parameters A and Q. As a result, this only changes the Kalman filter part in the post-intervention time steps from Algorithm 2 to Algorithm 6. The full description of $TASC_{pre}$ is provided in Algorithm 7.

Algorithm 7 $TASC_{pre}(Y; N_1)$, Parameter learning from pre-intervention only

---

Data: $y_{i,t}\ \forall (i,t) \in [0:n] \times [1:T_0]$ and $\forall (i,t) \in [1:n] \times [T_0+1:T]$

Result: $\hat{\theta} = \{A, H, Q, R, m_0, P_0\}, \hat{y}_{0,T_0+1}, \ldots, \hat{y}_{0,T}$

Learn $\theta^{N_1} \leftarrow EM_{pre}(Y_{pre}; N_1)$

for $k \leftarrow 1$ **to** T_0 **do**

  Update m_k, P_k via Algorithm 2 with θ^{N_1} `/* pre-intervention forward pass */`

end

for $k \leftarrow T_0 + 1$ **to** T **do**

  Update m_k, P_k via Algorithm 6 with θ^{N_1} `/* post-intervention forward pass */`

end

for $k \leftarrow T - 1$ **to** 0 **do**

  Update m_k^s, P_k^s via RTS Smoothing with $\theta^{(i-1)}$ `/* backward pass */`

end

Define $H = \begin{bmatrix} h_1^\top \\ H_2 \end{bmatrix}$

for $k \leftarrow T_0 + 1$ **to** T **do**

  Predict $\hat{y}_{0,t} \leftarrow h_1^\top m_t^s$ `/* Post-intervention target prediction */`

end

return $\theta^{N_1}, \hat{y}_{0,T_0+1}, \ldots, \hat{y}_{0,T}$

B.2 Continued Parameter Learning with Post-Intervention Data

One could continue to update the parameters based on the post-intervention data as well. In this case, we will have to augment the post-intervention target values in a realistic way, because the M-step (Algorithm 4) uses y_k to update the parameters. The most natural way is to use the most recent parameters to predict post-intervention target values, just the same as we do in the prediction phase of $TASC_{pre}$ (the last for-loop). Algorithm 8 shows this approach, $TASC_{full}$, where we run N_2 additional rounds of M-step to update the parameters, using both pre- and post-intervention data.

Algorithm 8 $TASC_{full}(Y; N_1, N_2)$, Continued Parameter Learning with Post-Intervention Data (the Full EM Algorithm)

Data: $y_{i,t}\ \forall (i,t) \in [0:n] \times [1:T_0]$ and $\forall (i,t) \in [1:n] \times [T_0+1:T]$
Result: $\theta = \{A, H, Q, R, m_0, P_0\}$
Learn $\theta^{N_1} \leftarrow EM_{pre}(Y_{pre}; N_1)$
for $i \leftarrow N_1 + 1$ **to** $N_1 + N_2$ **do**
  for $k \leftarrow 1$ **to** T_0 **do**
    Update m_k, P_k via Algorithm 2 with $\theta^{(i-1)}$ `/* pre-intervention forward pass */`
  end
  for $k \leftarrow T_0 + 1$ **to** T **do**
    Update m_k, P_k via Algorithm 6 with $\theta^{(i-1)}$ `/* post-intervention forward pass */`
  end
  Define $H = \begin{bmatrix} h_1^\top \\ H_2 \end{bmatrix}$
  for $k \leftarrow T - 1$ **to** 0 **do**
    Update m_k^s, P_k^s via RTS Smoothing with $\theta^{(i-1)}$ `/* backward pass */`
    Augment $\tilde{y}_k \leftarrow [h_1^\top m_k^s, y_{1,k}, \ldots, y_{n,k}]^\top$
  end
  Update $\theta^{(i)}$ via the M-step of EM with $\tilde{y}_k$
end
return $\theta^{(N_1+N_2)}$

C Advanced Model

Without loss of generality, we can add a constant state x_0 and let only x'_t part to change over time.

$$x'_t = Ax'_{t-1} + q_{t-1} \quad q_{t-1} \sim \mathcal{N}(0, Q), \tag{4}$$
$$x_t = x^* + x'_t \tag{5}$$
$$y_t = Hx_t + r_t \quad , r_t \sim \mathcal{N}(0, R). \tag{6}$$

The model parameters are $\theta = \{A, H, Q, R, m_0, P_0\}$, where $A \in \mathbb{R}^{d \times d}$, $H \in \mathbb{R}^{(n+1) \times d}$, $Q \in \mathbb{R}^{d \times d}$, $R \in \mathbb{R}^{(n+1)\times(n+1)}$, $m_0 \in \mathbb{R}^d$, $P_0 \in \mathbb{R}^{d \times d}$, and $x^* \in \mathbb{R}^d$. This model can be easily adopted with our algorithms with minimal modifications.

D Additional Experiments with Proposition 99 Data

In this section, we provide additional experiments we ran with Proposition 99 data that were not included in the main text due to the space limit.

First, we take a deeper dive into comparing RSC and our methods. Among the various SC methods we tested, $TASC$ and RSC explicitly filter the data matrix to be *exactly* low-rank before constructing the counterfactuals. To see how our *interpretable latent factor model* helps, we test the performance of two methods with varying d (approximate rank for RSC, hidden dimension for $TASC$). Figure 4 reports the post-intervention RMSE from the placebo test. The lowest error occurs with $TASC_{pre}$ at $d = 2$, followed by $TASC_{post}$, RSC. Both $TASC$ and RSC perform best at $d = 2$, but RSC's performance deteriorates rapidly as d increases, while $TASC$ remains relatively stable. Hence, $TASC$ may offer advantages in settings where the true value of d is difficult to estimate.

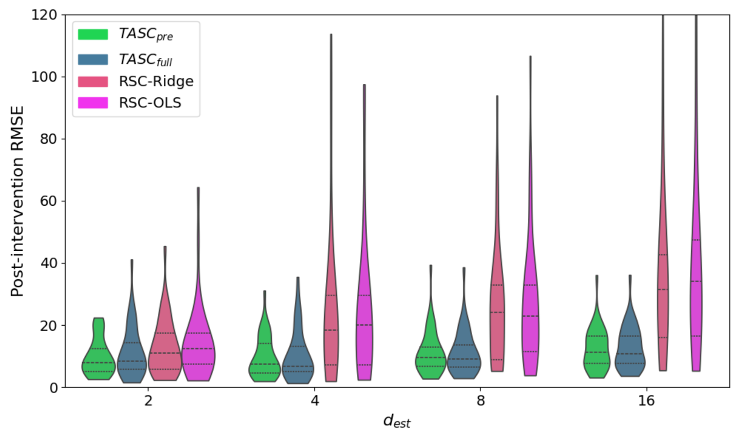

Figure 4: Post-intervention RMSE from placebo test, with varying d.

Next, we tested a boosting approach to help boost the accuracy of prediction. While running the experiments, we realized that the EM algorithm is extremely sensitive to the initialization of the parameters. This is especially important for larger d, as we need more rounds of EM updates with increased number of parameters to learn. This can be remedied by incorporating boosting rounds—the most naive approach can be iterating b rounds of EM and taking the median outcome as final estimate. Figure 5 shows the placebo test results for varying rounds of boosting, where we set $d = 8$.

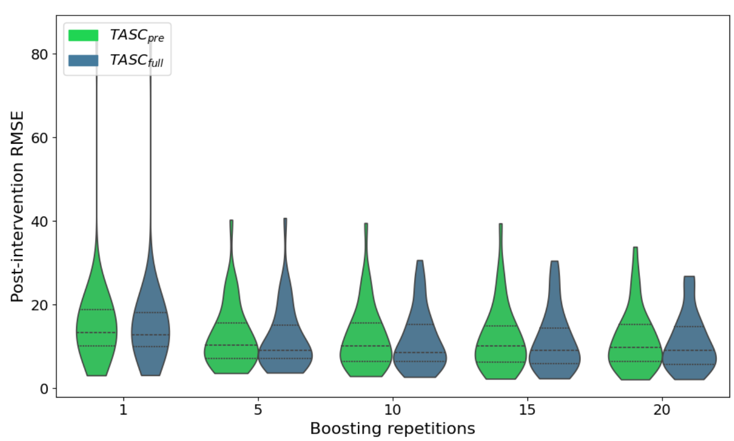

Figure 5: Post-intervention RMSE from placebo test with varying number of boosting rounds (1 means no boosting). We fixed $d = 8$ for all instances included in this plot.

Lastly, following the approach of [3], we plot the difference between the observed outcome and the predicted counterfactual for California, alongside those of the control states obtained from the placebo test. The first column of Figure 6 reports the gap in per-capita cigarette sales (in packs) for California and the 38 control states. Subsequent columns restrict the set of control states based on the relative quality of pre-intervention fit, measured by mean-squared error (MSE). The second column includes only control states whose pre-intervention MSE is no more than 10 times that of California, while the third and fourth columns apply stricter thresholds of 5 and 2 times, respectively.

Notably, the last row (RSC) retains the largest number of control states as stricter thresholds are imposed, indicating that the distribution of pre-intervention fit shows lower variance relative to the other methods. In contrast, the $TASC$ approaches (first two rows) retain substantially fewer control states under the most stringent threshold compared to SC or RSC. Across all specifications, California consistently displays the largest gap, while it is more apparent in the plots in the top right corner (stricter thresholds, $TASC$ methods).

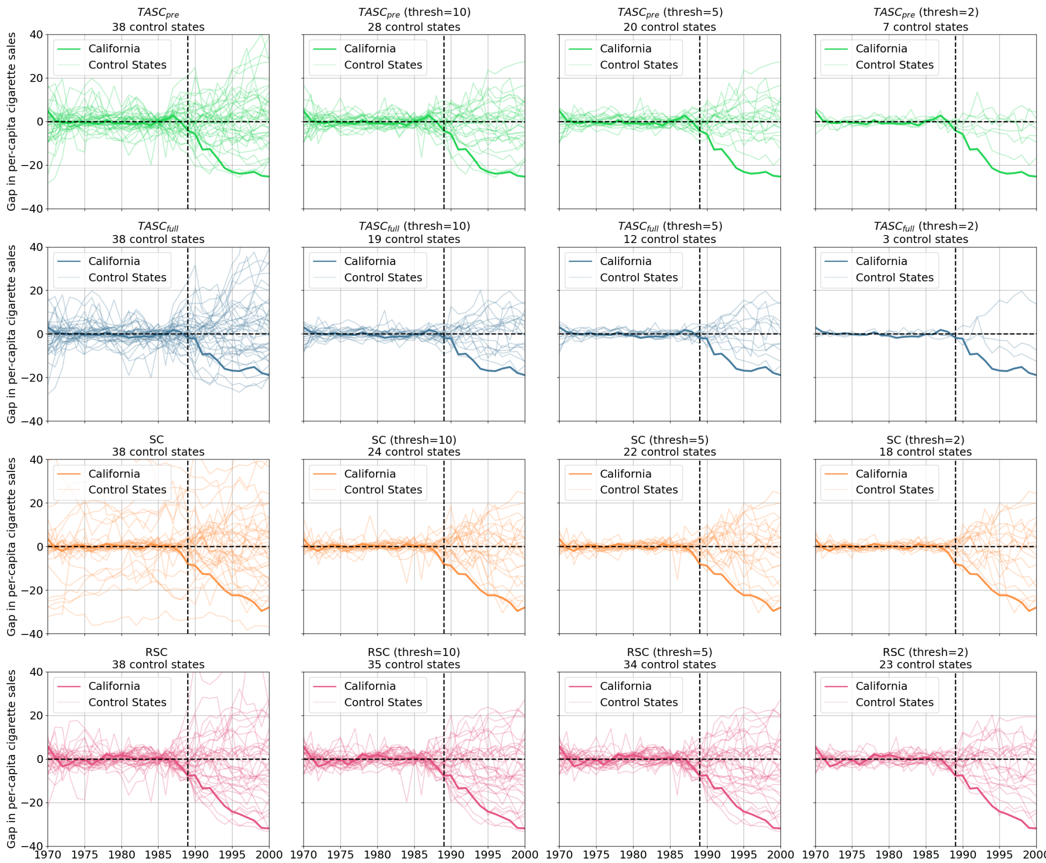

Figure 6: Gap in per-capita cigarette sales (in packs) between the observed outcome and the synthetic control (SC) predictions (comparable to Figures 4–7 in [3]). Each row corresponds to a different algorithm ($TASC_{pre}$, $TASC_{full}$, SC, and RSC, from top to bottom), and each column applies a different threshold for selecting control units (no threshold, at most 10 times California's pre-intervention error, 5 times, and 2 times, from left to right).