Cambridge Medium (2020), **37**, 1–39

CAMBRIDGE
UNIVERSITY PRESS

ARTICLE

Dynamic Synthetic Controls

Accounting for Varying Speeds in Comparative Case Studies

Jian Cao[*] and Thomas Chadefaux

Department of Political Science, Trinity College Dublin, 2 Clare Street, Dublin 2, Ireland

[*]Corresponding author. Email: caoj@tcd.ie

Abstract
Synthetic controls are widely used to estimate the causal effect of a treatment. However, they do not account for the different speeds at which units respond to changes. Reactions may be inelastic or "sticky" and thus slower due to varying regulatory, institutional, or political environments. We show that these different reaction speeds can lead to biased estimates of causal effects. We therefore introduce a dynamic synthetic control approach that accommodates varying speeds in time series, resulting in improved synthetic control estimates. We apply our method to re-estimate the effects of terrorism on income (Abadie and Gardeazabal 2003), tobacco laws on consumption (Abadie, Diamond, and Hainmueller 2010), and German reunification on GDP (Abadie, Diamond, and Hainmueller 2015). We also assess the method's performance using Monte-Carlo simulations. We find that it reduces errors in the estimates of true treatment effects by up to 70% compared to traditional synthetic controls, improving our ability to make robust inferences. An open-source R package, `dsc`, is made available for easy implementation.

Keywords: causal inference, time series, dynamic time warping, difference in differences, comparative case studies

1. Introduction

Social scientists often seek to estimate the causal effects of treatments, such as events or interventions, on various outcomes of interest. Researchers may, for example, investigate the consequences of introducing minimum wage legislation on unemployment, the influence of terrorist attacks on economic growth, or the effect of electoral redistricting on legislative behavior. Typically, making such causal inferences requires the construction of counterfactuals to infer what would have occurred in the absence of the treatment.

An important tool for constructing counterfactuals is the synthetic control method, “arguably the most important innovation in the policy evaluation literature in the last 15 years” (Athey and Imbens 2017). This method is designed to provide an accurate representation of the hypothetical outcome for the treated unit without the treatment. It involves combining non-treated units, which, when given appropriate weights, closely resemble the pre-treatment unit. This technique extends the difference-in-difference approach, with the advantage of generating a close match to the unit of interest, even when no single control unit would be appropriate on its own.

However, the synthetic control approach does not account for the potentially different speeds at which units react and adapt to changes. Changes in reactions to an event or a policy may be inelastic or “sticky” and therefore take longer in one unit than in another. They may also vary within one unit over time. A previously slow unit may now be fast; and speed may vary as a function of certain covariates.

Different units—states, cities, people—may adjust and react at varying speeds for many reasons. For instance, legal constraints in one state may slow down the speed at which economic decisions are made compared to other states. Öztekin and Flannery (2012), for example, find that the speed with which firms adjust their capital structure correlates with legal and financial norms. Additionally, the speed of convergence—the rate at which a country’s per capita income approaches its steady state—has also been shown to vary by country and over time (Rappaport 2000; Barro et al. 1991; Canarella, Pollard, et al. 2004). Similarly, the effect of a policy may be slower to fully manifest itself in one instance than in another. Thus, Alesina, Cohen, and Roubini (1992) find that the speed at which the deficit is adjusted in election years may be slower than in other years. Moreover, the impact of a shock on e.g. commodity prices, may be almost instantaneous in some markets but drag on in others.

Variations in speed—how quickly units react to changes—have important consequences for our ability to make inferences. We may, for example, conclude that a treatment had a large effect on treated unit $\mathbf{y}_1$ compared to untreated units $\mathbf{y}_2$ and $\mathbf{y}_3$. However, the observed difference in reaction may simply be due to delays in $\mathbf{y}_2$ and $\mathbf{y}_3$. Conversely, we may underestimate the effect on $\mathbf{y}_1$ if $\mathbf{y}_2$ reacts faster. Ignoring these differences in speed creates further problems for synthetic controls, as the donor[1] and the treated units may be out of synchronization, and what appears to be a good pre-treatment match may in fact be due to chance rather than a true alignment. Another problem is that the varying speeds may prevent us from finding a suitable synthetic control at all, as there may be no linear combination of the donors that allows for a close match to the treated unit.

To demonstrate the significance of these varying speeds for our ability to make inferences, suppose that we aim to estimate the causal effect of a policy on an outcome, e.g., the impact of cigarette taxes on consumption or the introduction of a minimum wage on employment. We have artificial data for three units (e.g., states) over time, as illustrated in Figure 1. During the observation period, one of the units ($\mathbf{y}_1$) receives a treatment, while the other two ($\mathbf{y}_2$ and $\mathbf{y}_3$) do not, allowing them to be used as potential donor units for counterfactual analysis. However, the researcher is not aware that the true effect of the treatment is zero.

To generate the synthetic unit, the standard synthetic control approach constructs a weighted average of the two available donor units $\mathbf{y}_2$ and $\mathbf{y}_3$ using weights $\mathbf{w}^* = (w^*_{\mathbf{y}_2}, w^*_{\mathbf{y}_3})$. $\mathbf{w}^*$ is selected such that the resulting synthetic control, which combines the non-treated units $\mathbf{Y}_{-1}$ as $\mathbf{Y}_{-1}\mathbf{w}^*$, closely matches the pre-treatment sequence of the treated unit $\mathbf{y}_1$.[2] Abadie and Gardeazabal (2003) demonstrate that this approach produces a plausible counterfactual for the treated unit.

In the example of Figure 1, however, selecting weights that generate a suitable counterfactual is not straightforward. Indeed, even in this simple case, there is no easily obtainable closed-form solution for finding $\mathbf{w}$. The standard linear approach results in the synthetic control depicted by the blue curve. This curve poorly matches both the pre- and post-treatment unit $\mathbf{y}_1$, and leads us to erroneously conclude that a treatment effect exists (the post-treatment difference between the blue and black curves) when, in reality, the true treatment effect is zero. Adding nonlinear terms or lags does not improve the situation much either (see Figure A1 in Appendix 1, which incorporates multiple lags and polynomial terms).

The approach's challenge to generate a satisfactory approximation of the treated unit using untreated units stems from the challenge of accounting for varying speeds. In this case, $\mathbf{y}_2$, $\mathbf{y}_3$, and $\mathbf{y}_1$ all exhibit cycles that unfold at different rates. While all three units display the same patterns, $\mathbf{y}_2$'s cycle is longer (i.e., slower) than $\mathbf{y}_1$'s, and $\mathbf{y}_3$ is faster than $\mathbf{y}_1$. Additionally, the differences in speed fluctuate within each unit over time, so that a single speed adjustment parameter is insufficient.

[1] In synthetic control, the untreated units are refereed as donors.

[2] In other words, we choose $\mathbf{w}^*$ to satisfy $\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}^*} \|\mathbf{y}_1 - \mathbf{Y}_{-1}\mathbf{w}^*\|_2$. For simplicity, we abstract away any predictors.

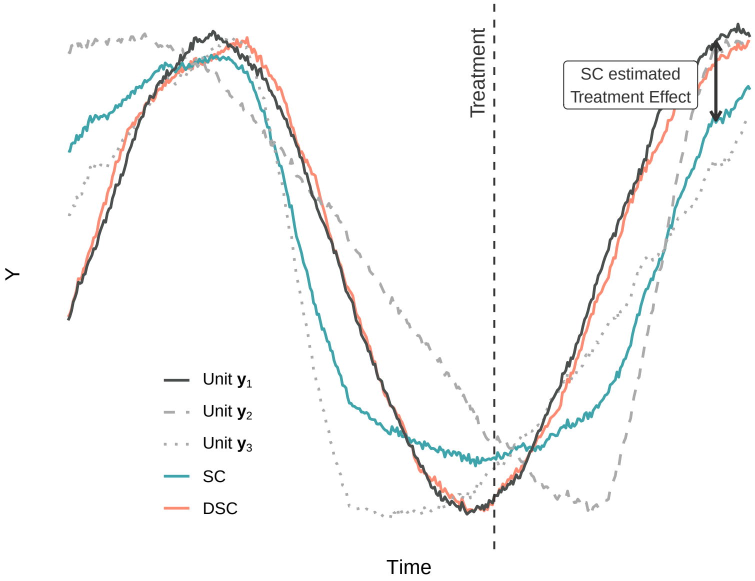

Figure 1. The challenge of varying speeds. Consider a researcher aiming to quantify the effect of a treatment on unit $\mathbf{y}_1$. Unbeknownst to them, no treatment effect actually exists. Employing conventional Synthetic Control (SC) methods with donor units $\mathbf{y}_2$ (slow) and $\mathbf{y}_3$ (fast), they obtain a post-treatment estimate (blue curve) that diverges markedly from the true outcome (black curve), leading to a significant bias in the estimated treatment effect. In contrast, Dynamic Synthetic Controls (DSC), as elaborated below, produce a synthetic control (red curve) that more closely approximates the truth.

Lastly, variations in speed might depend on unobservable variables or endogenous regressors, further complicating the process of correcting speeds.

In this paper, we introduce a new method, Dynamic Synthetic Controls (DSC), which accounts for varying speeds within and across units. This approach operates by learning the differences in speed between the series during the pre-treatment period. We do this by calculating a Dynamic Time Warping (DTW) path between them. The warping path subsequently provides a measure of speed differentials across units and over time. We then use this warping path to warp the post-treatment donor units to align their speed with that of the treated unit. As a result, we can assess the treatment effect while controlling for the inherent varying speeds. As an illustration of the method's capabilities, the red curve in Figure 1 shows the outcome of applying DSC to this simple case. We observe that warping enables us to more closely match the post-treatment unit compared to a standard synthetic control. Importantly, the estimator only removes speed differences that originate from the pre-treatment period; it does not remove speed differences that are caused by the treatment. This, in turn, allows for a more precise inference of the treatment effect's magnitude.

The remainder of this paper demonstrates that this result extends beyond a simple example. In fact, we show empirically that Dynamic Synthetic Controls serve as a more efficient estimator of the treatment effect compared to standard synthetic controls. We substantiate these findings by replicating leading empirical work in this field and illustrating how the results can be improved using Dynamic Synthetic Control. Additionally, we generalize our results through a Monte-Carlo simulation.

2. Synthetic Controls and the Problem of Speed

Many of the questions of interest to social scientists revolve around estimating the effect of treatments such as an event or an intervention (e.g., Card and Krueger 2000; Chattopadhyay and Duflo 2004; Di Tella and Schargrodsky 2004; Brady and McNulty 2011). Difference-in-differences, for example, leverages similarities between a treated unit and an untreated one to deduce the treatment's effect. However, the method potentially suffers from biased control selection and imprecise case similarity—sometimes lacking a suitable comparison altogether.

The synthetic control approach, pioneered by Abadie and Gardeazabal (2003) and refined in Abadie, Diamond, and Hainmueller (2010, 2015), aims to address such limitations by combining a basket of control cases to mimic the pre-treatment scenario for the treated unit. The enables a more robust counterfactual study of the treatment's impact (see (Abadie 2021) for a review. A refresher of the method is in Appendix 2).

However, in a wide class of situations, there is in fact no easy way to generate a good counterfactual using standard synthetic control method. Suppose that we observe a time series $y_{1,t}$ exposed to a treatment at time T. We also observe a basket of time series $y_{j,t}, j \in (2, 3, \ldots, J+1)$ to be used as donors. All time series are of length N. Assuming that the target time series $y_{1,t}$ depends not only on current $y_{j,t}$, but also on lags $y_{j,t-l}, l \in (1, 2, \ldots, t-1)$,

then the model of interest for the behavior of $\mathbf{y}_1$ over time is

$$y_{1,t} = \sum_{j=2}^{J+1} [w_j (y_{j,t} + \beta_{1,j,t} y_{j,t-1} + \beta_{2,j,t} y_{j,t-2} + \ldots \beta_{t-1,j,t} y_{j,1})] + \epsilon_t, \tag{1}$$

where $\beta_{l,j,t}$ is a time-dependent coefficient for the lag term $y_{j,t-l}$, l refers to the order of the lag, j to the donor unit, and t to the time. w_j is a constant weight for donor $\mathbf{y}_j$ and its lags, and ϵ_t is the error term. We assume that all classical assumptions about ϵ_t apply. This model explores the speed problem by allowing a varying number of lags of $y_{j,t}$ to influence the target time series $y_{1,t}$. The higher order of lags that have non-zero coefficients, the "slower" $y_{1,t}$ is relative to $y_{j,t}$—i.e., the more the past drags on.

Consider now the standard synthetic control approach. As it only includes the current donor time series $y_{j,t}$ but not the lags $y_{j,t-l}$, the model used in estimating the weights $\mathbf{w}_{sc}$ is not the one of equation 1, but rather:

$$y_{1,t} = \sum_{j=2}^{J+2} (w_j y_{j,t}) + \eta_t = \mathbf{Y}_{-1,t}\mathbf{w}_{sc} + \eta_t,$$

and the new error term η_t is

$$\eta_t = \mathbf{Y}_{-1,L}\boldsymbol{\beta}_L\mathbf{w} + \epsilon_t$$

where $\mathbf{Y}_{-1,L}$ are lag terms, $\boldsymbol{\beta}_L$ are lag coefficients. Clearly, ignoring the lagged effects of the donor time series results in omitted variable bias. This subsequently leads to a biased estimate of the treatment effect and inflated standard errors (see Appendix 3 for a proof).

Various extensions of the synthetic control method have been introduced to tackle the problems of poor pre-treatment fits, but none are suitable for the varying speed problem. For example, Ben-Michael, Feller, and Rothstein (2022) examine the synthetic control method in scenarios where policies are implemented by different units at distinct times. Other than synthetic control methods, Goodman-Bacon (2021) discusses how improved difference-in-difference estimators address the bias from time-varying treatment effects. While many newly-developed methods (Ben-Michael, Feller, and Rothstein 2022; Ferman, Pinto, and Possebom 2020; Goodman-Bacon 2021) strive to enhance causal inference outcomes when the data requirements outlined in Abadie and Cattaneo (2021) are not met, none specifically address the varying speed problem which involves estimating the *JNN* lag coefficients.[3]

In this paper, we address the speed problem by relying on Dynamic Time Warping (DTW), a non-parametric method derived from speech recognition (Vintsyuk 1968), which allows us to recover speed differences between sequences. The DTW algorithm is widely used to find the optimal

[3] (*J* donor time series) × (*N* time periods) × (*N*/2 backward lags + *N*/2 forward lags). Please see Appendix 4 for a discussion of forward lags.

alignment between two time series that may vary in speed. It obtains the alignment by warping the sequences in the time dimension, such that the Euclidean distance between the warped and the target sequences is minimized. Determining the best way to warp the series is equivalent to estimating the lag coefficient βs.

After estimating the lag coefficient βs, we combine the lag terms into a single time series

$$y_{j,t}^{w} = \sum_{l=t-N}^{t-1} \hat{\beta}_{l,j,t} y_{j,t-l},$$

which we refer to as a "warped" time series. Then when applying synthetic control, we replace the original donor time series $\mathbf{y}_j$ with the warped time series $\mathbf{y}_j^w$ to mitigate the impact of the speed problem. We provide details of this process below.

3. Accounting for Speed: Dynamic Synthetic Controls

The speed issue arises from two possible sources: those caused by the treatment itself and those resulting from inherent speed differences (see Figure 1 and the previous section). Here, we introduce a method designed to mitigate the inherent speed differences between treated and untreated units, while maintaining the effects induced by the treatment—whether in level or speed. As a result, we can attribute any post-treatment differences between the two groups exclusively to the treatment. This approach eliminates the influence of pre-existing speed disparities in the time series.

The Dynamic Synthetic Control (DSC) method introduced below addresses the issue of varying speeds across time and units. It begins by warping the untreated units to minimize the speed differences between them and the treated unit. The algorithm extracts information on speed differences from the pre-treatment period, and adjusts the speed of the entire donor time series to align with the treated time series.[4] This indirect warping approach allows the remaining difference between the treated and donor units to be attributed exclusively to treatment differences rather than inherent speed differences. After adjusting the series' speeds, the standard synthetic control method is applied.

The warping process comprises three steps (see Figure 2 and Algorithm A1 in Appendix 5). First, we match the pre-treatment parts of the target (black) and the donor (red) time series; second, we match the post-treatment and the pre-treatment parts of the donor time series; finally, we combine

[4] Importantly, we avoid inferring speed differences from the post-treatment period, as it could conflate the effects of inherent speed differences and treatment, leading to biased estimates.

the time alignments obtained in the previous two steps and create a warped donor time series (blue) that minimizes the inherent speed difference between itself and the target time series (black). We now discuss each step in detail.

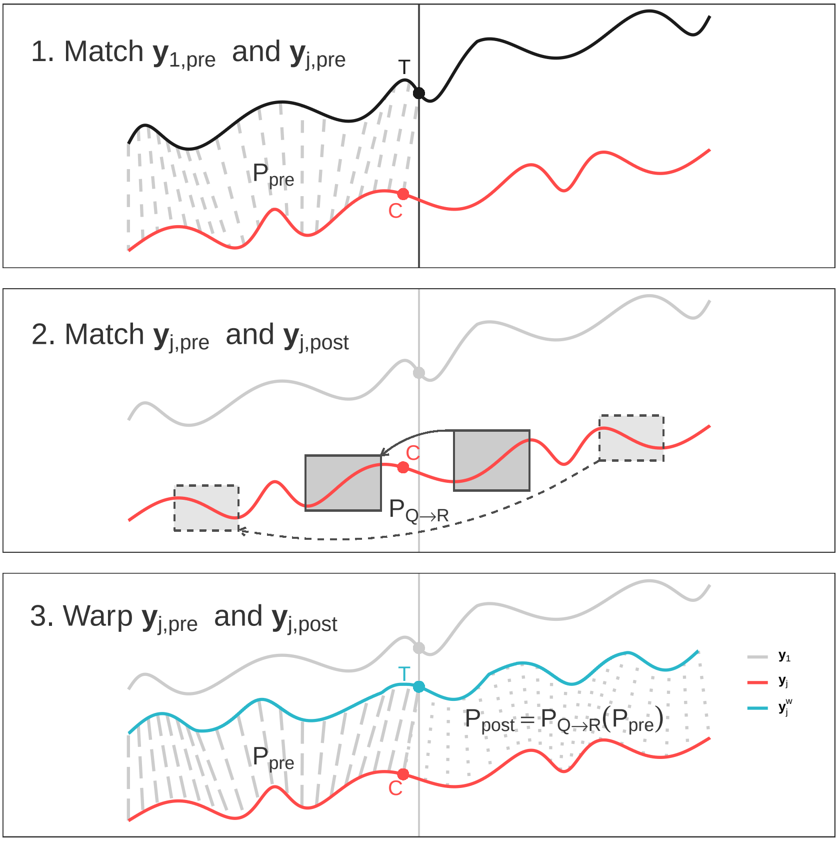
1. Match $\mathbf{y}_{1,pre}$ and $\mathbf{y}_{j,pre}$
T
P_{pre}
C
2. Match $\mathbf{y}_{j,pre}$ and $\mathbf{y}_{j,post}$
C
$\mathrm{P}_{Q\to R}$
3. Warp $\mathbf{y}_{j,pre}$ and $\mathbf{y}_{j,post}$
T
$\mathrm{P}_{post} = \mathrm{P}_{Q\to R}(\mathrm{P}_{pre})$
P_{pre}
C
$\mathbf{y}_1$
$\mathbf{y}_j$
$\mathbf{y}_j^w$

Figure 2. Dynamic Synthetic Control (DSC). The warping process of the DSC algorithm operates in three key steps. First, it matches the pre-treatment segments of $\mathbf{y}_j$ and $\mathbf{y}_1$ to derive the warping path $\mathbf{P}_{pre}$. Second, it aligns the pre- and post-treatment segments of $\mathbf{y}_j$, yielding $\mathbf{P}_{Q\to R}$. Finally, $\mathbf{y}_j$ is warped using both $\mathbf{P}_{pre}$ and $\mathbf{P}_{post}$ to produce the time-warped series $\mathbf{y}_j^w$.

Step 1. Matching Pre-treatment Time Series

The DSC algorithm first estimates the speed relationship between the pre-treatment segments of the target time series $\mathbf{y}_1$ and the donor time series $\mathbf{y}_j$. This is achieved by employing the Dynamic Time Warping method to align the pre-treatment portion of $\mathbf{y}_1$ with $\mathbf{y}_j$ and storing the warping path matrix.

The warping path matrix stores the results of Dynamic Time Warping (DTW) alignments and is formally defined as follows:

$$\mathbf{P}_j = [p_{j,\upsilon,t}], \quad \upsilon, t \in [1, N]$$

$$p_{j,\upsilon,t} = \begin{cases} 1 & \text{if } y_{j,\upsilon} \text{ matches } y_{1,t} \\ 0 & \text{otherwise} \end{cases}$$

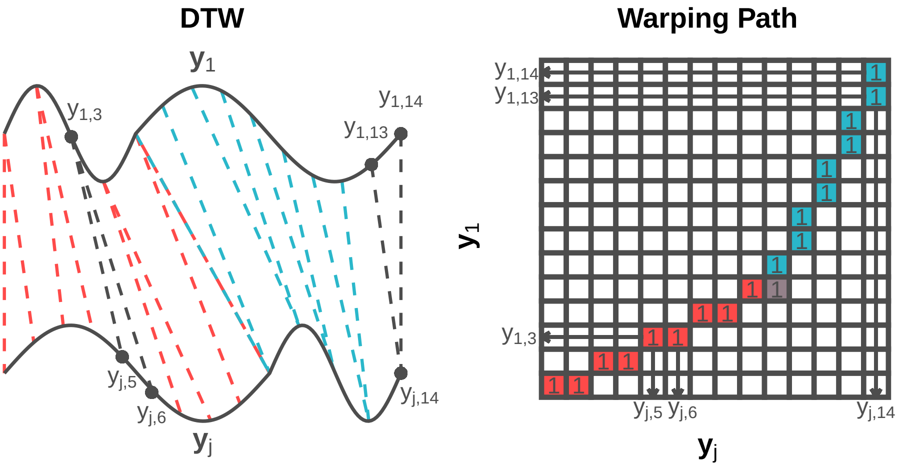

Figure 3. Warping path. The left figure shows data points matched in DTW, connected by dashed lines. The right figure displays the corresponding warping path matrix, where only matched pairs (ones) are shown. The time series $\mathbf{y}_j$ initially progresses at a rate 2× slower (indicated in red) than $\mathbf{y}_1$ but later becomes 2× faster (in blue) than $\mathbf{y}_1$.

As illustrated in Figure 3, the points of $\mathbf{y}_1$ are matched to the points of $\mathbf{y}_j$ by DTW in a manner that minimizes the total distance between the matched points.[5] The warping path matrix $\mathbf{P}_j$ captures the speed difference between $\mathbf{y}_1$ and $\mathbf{y}_j$.

We first match the pre-treatment target time series $\mathbf{y}_{1,1:T}$ to the donor time series $\mathbf{y}_j$ (see the first part in Figure 2). The point $y_{1,T}$ is matched to $y_{j,C}$.[6] Let $\mathbf{y}_{j,pre} = \mathbf{y}_{j,1:C}$ denote the pre-treatment part of $\mathbf{y}_j$ and $\mathbf{y}_{j,post} = \mathbf{y}_{j,C:N}$ the post-treatment portion. Similarly, we define $\mathbf{y}_{1,pre} = \mathbf{y}_{1,1:T}$ and $\mathbf{y}_{1,post} = \mathbf{y}_{1,T:N}$. Using Dynamic Time Warping, we obtain a warping path from $\mathbf{y}_{j,pre} \to \mathbf{y}_{1,pre}$, which is stored in a $C \times T$ matrix $\mathbf{P}_{j,pre}$.

The warping path $\mathbf{P}_{j,pre}$ is used in step three to warp the time axis of $\mathbf{y}_{j,pre}$ with the goal of

[5] Formally, $\mathbf{P}_j = \text{argmin}_{\mathbf{P}_j} \left(\sum_{t=1}^{N} \sum_{\upsilon=1}^{N} \frac{p_{j,\upsilon,t}|y_{1,t}-y_{j,\upsilon}|}{\sum_{\upsilon=1}^{N} p_{j,\upsilon,t}} \right)$.

[6] To ensure optimal matching, we do not impose an end rule. Consequently, the matched time series $\mathbf{y}_{1,1:T}$ and $\mathbf{y}_{j,1:C}$ may have different lengths.

minimizing the speed difference between $\mathrm{y}_{j,pre}$ and $\mathrm{y}_{1,pre}$. Additionally, it is employed along with the warping path derived from step two to adjust the speed of $\mathrm{y}_{j,post}$.

Step 2. Matching Pre- and Post-treatment Donor Time Series

Learning the warping path between the post-treatment segments $\mathrm{y}_{j,post}$ and $\mathrm{y}_{1,post}$ is more challenging. We cannot directly align the two sequences, because their differences are due not only to their different speeds, but also to the treatment effect. Aligning them would artificially remove that treatment effect. So the post-treatment warping path $\mathbf{P}_{j,post}$, i.e., the inherent differences in speed, must be learned from the pre-treatment path $\mathbf{P}_{j,pre}$.

To extract similar short-term sequences in $\mathrm{y}_{j,post}$ and $\mathrm{y}_{j,pre}$ and therefore infer the warping path $\mathbf{P}_{j,post}$ from $\mathbf{P}_{j,pre}$, we use a double-sliding window approach.[7] In essence, the double-sliding window serves as a dynamic "lens"—the first window scans through the post-treatment time series to identify segments requiring alignment, while the second window sifts through the pre-treatment time series to find the most similar segments, which facilitates optimal matching and warping of patterns between the two time series. As illustrated in the second part of Figure 2, we slide a target window Q_u in $\mathrm{y}_{j,post}$ and a reference window R_i in $\mathrm{y}_{j,pre}$. For each short-term sequence Q_u in $\mathrm{y}_{j,post}$, we find the best-matching sequence R^* from $\mathrm{y}_{j,pre}$, and record the warping path $Q_u \to R^*$ as $\mathbf{P}_{j,Q_u \to R^*}$. We also extract the corresponding pre-treatment warping path $\mathbf{P}_{j,R^*} : R^* \to \mathrm{y}_1$ from $\mathbf{P}_{j,pre}$. Next, $\mathbf{P}_{j,R^*}$ is adjusted based on $\mathbf{P}_{j,Q_u \to R^*}$ to account for the differences between Q_u and R^*. The resulting adjusted warping path is $\mathbf{P}_{j,Q_u}$. Once the sliding target window Q_u reaches the right boundary $y_{j,N}$, we merge all the resulting warping paths $\mathbf{P}_{j,Q_u}$ to obtain the final warping path $\mathbf{P}_{j,post}$ for the post treatment donor time series.[8]

Step 3. Warping Donor Time Series

In the third step, the DSC algorithm uses the previously estimated warping path $\mathbf{P}_{j,pre}$ to warp the pre-treatment donor series $\mathrm{y}_{j,pre}$. Similarly, it uses $\mathbf{P}_{j,post}$ from the second step to warp $\mathrm{y}_{j,post}$.[9] The

[7] Details on the double-sliding window approach can be found in Appendix 6.

[8] To ensure that the resulting warping path $\mathbf{P}_{j,post}$ has the best accuracy, a threshold θ is applied to the DTW distances to filter out any less desired matches. If for a target window Q_u, we cannot find any matches that meet the standard θ, the target window will be unchanged, i.e. no warping will be given.

[9] To warp the time series using a warping path matrix $\mathbf{P}_j$, we first obtain estimates of lag coefficients $\hat{\beta}_{\upsilon-t,j,t} = \frac{p_{j,\upsilon,t}}{\sum_{\upsilon=1}^{N} p_{j,\upsilon,t}}$, then combine the lag terms and obtain the warped time series $y_{j,t}^{w} = \sum_{l=t-N}^{t-1} \hat{\beta}_{l,j,t} y_{j,t-l}$.

resulting time series are combined into a single warped donor time series $\mathbf{y}_j^w$:

$$\begin{aligned}\mathbf{y}_j^w &= [\mathbf{y}_{j,pre}^w, \mathbf{y}_{j,post}^w] \\ &= [\mathbf{P}_{j,pre}(\mathbf{y}_{j,pre}), \mathbf{P}_{j,post}(\mathbf{y}_{j,post})]\end{aligned}$$

After being warped, the inherent speed differences between $\mathbf{y}_j^w$ and $\mathbf{y}_1$ are minimized while the differences caused by the intervention are unchanged.

Finally, using the warped donor time series $\mathbf{y}_j^w, j \in (2, 3, \ldots, J+1)$, we apply the synthetic control method to construct a counterfactual of $\mathbf{y}_1$ to minimize the impact of the speed problem.

All methods discussed in this paper have been implemented in an accompanying R package.[10]

4. Evaluating the Method

We showcase the advantages of our proposed method using two approaches. First, we create synthetic data using a Monte Carlo simulation. This lets us design and know the treatment effect, in contrast to real-world data where the true treatment effect is never known. This allows us to show that, across a wide range of parameters, our estimate of the treatment effect is more efficient than the one obtained using the standard synthetic control approach.

Second, we apply the dynamic synthetic control method to data from three seminal articles on synthetic controls: Abadie and Gardeazabal (2003)'s data on the effect of terrorist attacks on GDP; Abadie, Diamond, and Hainmueller (2010)'s data on changes in tobacco consumption in California; and Abadie, Diamond, and Hainmueller (2015)'s study of the economic impact of the 1990 German reunification. In each case, we demonstrate that our estimates of the placebos are more efficient, such that the statistical test of the effect of the treatment has greater power.

In each case, we make two main arguments. First, we argue that our method generates a better counterfactual than the standard synthetic control. Second, we demonstrate that our method enhances precision by reducing uncertainty in the treatment effect estimate. A key issue in synthetic control is the possibility that observed effects occur by chance. Existing literature often shows that while synthetic controls for untreated units (placebo units) generally yield less extreme results than the treated unit, the confidence intervals remain wide. Our Dynamic Synthetic Control method

[10] The R package can be found in Appendix 10

effectively narrows these intervals, thereby improving estimator precision.

4.1 Monte-Carlo simulation

We first use a Monte Carlo simulation to replicate the types of empirical challenges faced with real data. We generate hundreds of artificial panel datasets, each containing time series for ten units observed over 100 periods. In each sample of ten series, one time series is designated as the "treated" unit, while the remaining nine serve as the donor pool to construct the synthetic control (i.e., $J = 9$, $N = 100$).[11]

All ten time series follow a common Autoregressive integrated moving average (ARIMA) process but exhibit differing speeds. A time series' speed is either random or a function of the time series direction (increasing or decreasing). The idea behind it is to capture the possibility that speed may vary as a function of the direction of the underlying series. Economic crashes (i.e. a decreasing series), for example, may unfold faster than recoveries (Reinhart and Rogoff 2014). A dataset-specific parameter $\psi \in (0, 1)$, determines the extent to which this occurs. For instance, $\psi = 0$ implies that the speed will be entirely governed by a random normal noise. Conversely, if $\psi = 1$, the speed will completely depend on the direction of the time series.

For each dataset, we implement Dynamic Synthetic Control (DSC) and standard Synthetic Control (SC) to construct counterfactuals for the treated unit. A key advantage of the Monte Carlo simulation, as opposed to real observational data, is that we know the true treatment effect and thus the true counterfactual. Consequently, we can assess how well counterfactuals formed using different methods approximate the post-treatment series. In accordance with common practice in the literature, we evaluate the performance of each method over the 10 periods following the treatment, i.e., from $t = 61$ to $t = 70$.

To evaluate the quality of the synthetic control generated by each method, we compute the 10-period post-treatment Mean Squared Error (MSE) for the estimated treatment effects. Specifically, the MSE for dataset d is defined as: $\text{MSE}_d = \frac{1}{10}\sum_{t=61}^{70}(\hat{\tau}_{t,d} - \tau_t)^2$, where $\hat{\tau}_{t,d}$ denotes the estimated treatment effects using one of the two synthetic control methods. A small MSE indicates that the estimated treatment effect closely fits the true effect imposed during the data generation process. For each simulation, we calculate the MSE for the standard Synthetic Control approach—MSE_{SC}—

[11] Detailed data generation method is shown in Appendix 8.

and the MSE for our Dynamic Synthetic control approach—MSE_{DSC}. The log ratio of these two values yields a measure r representing the relative performance of each method, that is, $r = \log(\text{MSE}_{DSC}/\text{MSE}_{SC})$. Negative log ratios suggest superior performance of our method via lower MSE. We compute the log ratio r for each simulated data set and conduct a t-test on the resulting 100 log ratios to assess whether the ratio is significantly different from 0 ($p < 0.01$).

The estimated treatment effects for all data sets are illustrated in Figure 4. We see that, in the ten periods following the treatment, the average estimated treatment effects from the two methods are close to the true effects. However, the new method, DSC, produces a significantly smaller error area than the standard method. Moreover, our approach outperforms the standard method 77% of the time (Wilcox test < 0.001). These results suggest a strong expected benefit to using this method and are in line with our theory: Section 2 shows that the estimated treatment effect from SC is potentially biased and has a larger variance. In the Monte Carlo study, we do not directly observe bias because the biases around the true values tend to average out to zero. But it is clear in the Monte Carlo results that the SC estimator has a higher variance, and hence that the DSC estimator of the treatment effect is more efficient.

Finally, we also show that the results hold for all values of ψ—the parameter which controls the extent to which speed varies as a function of the shape of the series (see the subfigure in Figure 4), although larger ψs lead to more significance (more negative t value). When $\psi = 1$, the average log ratio of Mean Squared Errors (MSEs) is $\bar{r} = -1.26$. This value indicates that, on average, the Dynamic Synthetic Control (DSC) method reduces the MSE of the estimated treatment effect by 71.63% when compared to the standard Synthetic Control (SC) method ($1 - e^{-1.26} \approx 71.63\%$).

4.2 Re-evaluating Empirical Findings

We now show that these results are not limited to artificial data but extend to real-world empirical data. Specifically, we apply our method to three seminal articles in the synthetic control literature: Abadie and Gardeazabal (2003)'s analysis of the economic costs of terrorism in the Basque country; Abadie, Diamond, and Hainmueller (2010)'s assessment of the effect of Proposition 99—a large-scale tobacco control program implemented in California in 1988; and Abadie, Diamond, and Hainmueller (2015)'s evaluation of the economic impact of the 1990 German reunification on West Germany.

One challenge, of course, is that we do not know the true treatment effect for the main unit of

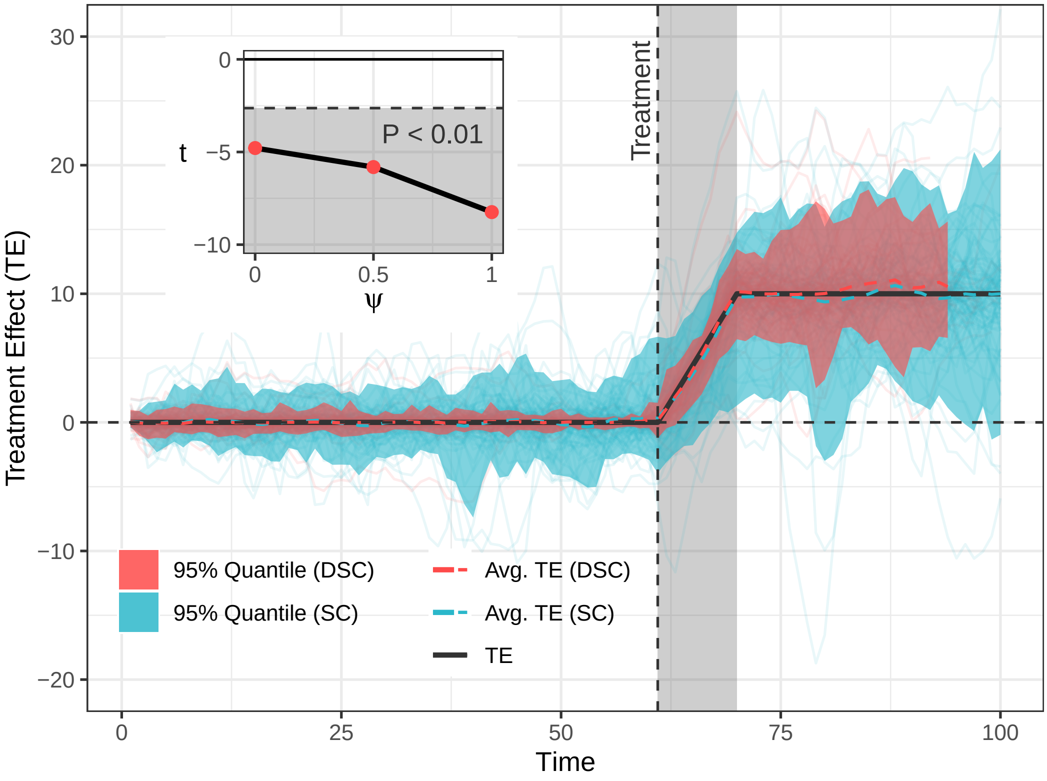

Figure 4. Results from the simulation study with 95% confidence intervals. The main graph showcases results drawn from Monte Carlo simulations where $\psi = 1$. The grey shaded region delineates the period over which performance is estimated. Red and blue lines represent the distribution of estimated treatment effects for the Dynamic Synthetic Control (DSC) and Synthetic Control (SC) methods, respectively. The true treatment effect is in black. An inset in the top-left corner demonstrates that larger ψ values lead to improved performance—as evidenced by more negative t-values.

interest. This effect must be estimated and, unlike in the Monte-Carlo study, there is no way to argue that a particular result is more accurate or less biased than another. However, we do know that there is no treatment effect in the other (non-treated) units. Thus, we can assess the performance of the estimators by comparing how well their respective synthetic controls approximate the true post-treatment period for non-treated units. For example, we cannot know the true effect of the German reunification on West Germany's GDP, but we do know that there was no treatment in Canada, and can thus attempt to match Canada's post-1990 trajectory—as well as each of the untreated units.[12] Due to the limited set of untreated units, however, this is not enough to yield a sufficiently large set of estimates for comparison. We therefore use jacknife resampling to generate more datasets in which we randomly remove one of the countries.

[12] That is, we estimate the 'treatment effect' for Canada using Japan, France, the USA, etc.; we estimate the treatment effect for France using Japan, Canada, Japan, and so on for all untreated units.

We now review each of the three datasets and the results obtained using each method.

Terrorism and GDP per capita in the Basque country. Abadie and Gardeazabal (2003) find that the outbreak of terrorism in the late 1960's significantly affected per capita GDP in the Basque Country. When compared to a synthetic control region without terrorism, the Basque Country's GDP declined by about 10 percentage points.

Our estimate of the effect on the Basque country is similar to the standard synthetic control estimate. However, when we build synthetic controls for the untreated units themselves, we find that they are closer to the true trajectory than is the standard synthetic control. Figure 5 (top) displays the distribution of our estimated treatment effects for all untreated units. Since these units did not receive a treatment, our synthetic control should ideally be as close as possible to the post-treatment values of the time series. In other words, the average difference between the post-treatment series and the synthetic control should deviate as little as possible from zero. Visually, we observe that the band for our approach is narrower than the one for standard synthetic control.

However, this plot does not capture the full extent of the true improvement, as we should be comparing the pairwise performance of each algorithm (instead of the pooled comparison shown here). To more formally demonstrate the improvement, we calculate the Mean Squared Error for each synthetic control compared to the unit of interest (in the same way as in the Monte-Carlo simulation above). We then calculate a t-test of $\log(\text{MSE}_{DSC}/\text{MSE}_{SC})$. A negative value indicates that our MSE is smaller than the one obtained with the standard approach. We find that this is indeed the case. The log ratio of the MSEs is significantly less than zero ($t = -7.6, p < 0.0001$).

In terms of our ability to make inferences, the observed reduction in Mean Squared Error (MSE) when employing the Dynamic Synthetic Control (DSC) method suggests its superior efficiency compared to traditional Synthetic Control (SC) methods. In short, our DSC approach more closely approximates the true treatment effects and therefore enhances the robustness and reliability of our causal inference.

The Effects of Proposition 99 on Tobacco Sales in California. Abadie, Diamond, and Hainmueller (2010) study the impact of Proposition 99, a large-scale tobacco control program implemented in California in 1988. They show that by 2000, California's per-capita cigarette sales were 26 packs fewer than would have been expected without Proposition 99.

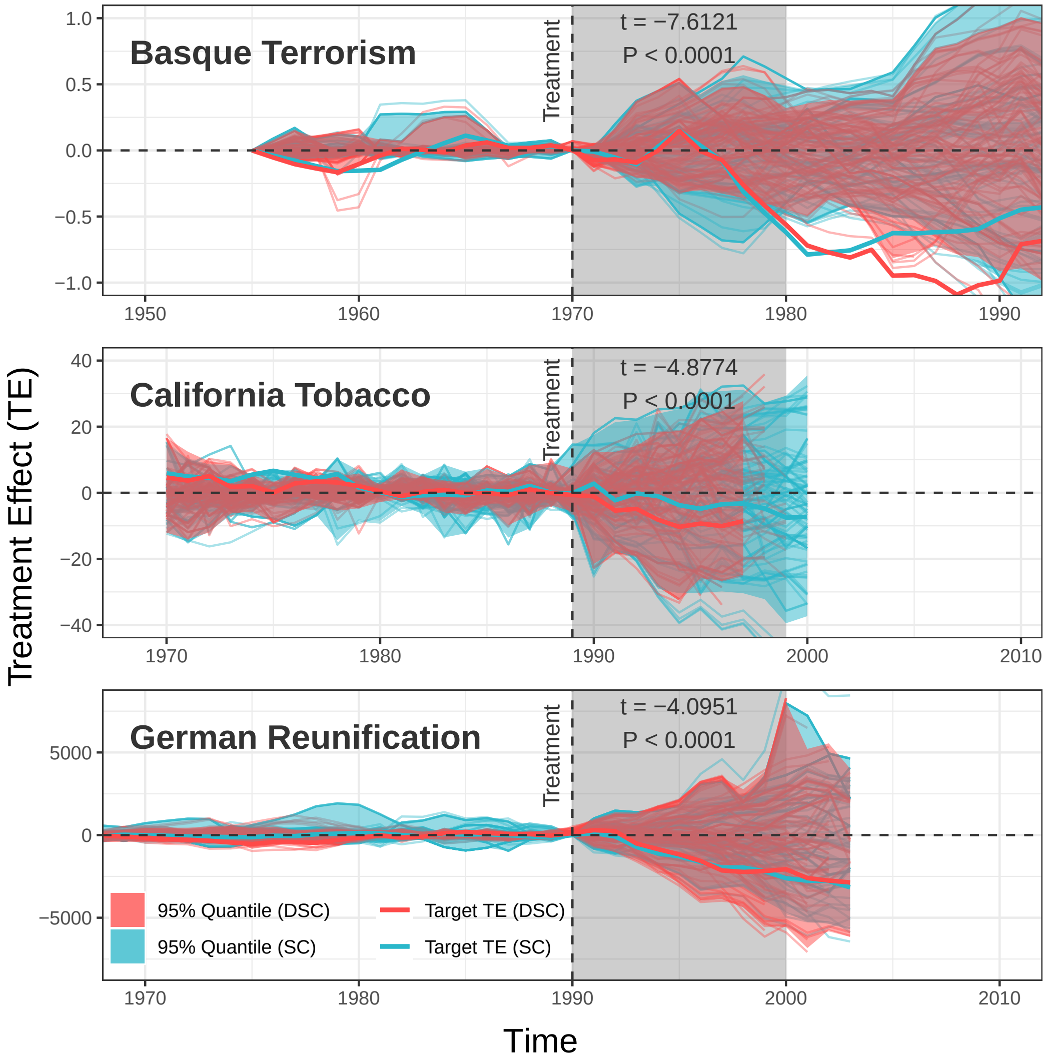

Figure 5. Placebo tests, real data. We revisited the placebo tests reported in Abadie and Gardeazabal (2003), Abadie, Diamond, and Hainmueller (2010) and Abadie, Diamond, and Hainmueller (2015). The plots report the placebo tests for each of these studies, using standard synthetic controls (blue) and dynamic synthetic control (red). In addition, the estimated treatment effects for the treated units—Basque Country, California State, and West Germany—are shown as thick, brighter lines. For each study, find that our placebo estimates exhibit smaller variance than those using standard synthetic controls, which do not account for variations in speed.

Our analysis suggests that the effect might in fact be greater, with an estimated reduction of about 31 packs in cigarette sales. Although it is impossible to definitively determine which estimate is more accurate, as the true treatment effect remains unknown, we do find that our estimates of the post-treatment behavior of states other than California is closer to the true path than is the standard synthetic control, with smaller MSEs ($t = -4.9, p < 0.001$).[13]

[13] Note that the synthetic controls generated by our method are shorter than those of the standard synthetic control. This

The Effects of the German Reunification on West Germany's GDP per capita. Finally, Abadie, Diamond, and Hainmueller (2015) seek to estimate the economic impact of the 1990 German reunification on West Germany. They find that the per-capita GDP of West Germany was reduced by on average about 1,600 USD per year over the 1990-2003 period, which is approximately 8% of the 1990 level. In 2003, the last year of their study period, The estimated per-capita GDP in the synthetic data is about 12% higher than in the real data.

In this replication, our estimate of the treatment effect on West Germany is similar to the one of Abadie and Gardeazabal (2003), and visually the 95% quantile areas of the estimated treatment effect of two synthetic control methods on the untreated countries are also close. But in the pair-wise comparison, we find that the DSC method generally exhibits a reduced Mean Squared Error (MSE) in estimating treatment effects for control countries, compared to the standard Synthetic Control approach ($t = -4.1, p < 0.001$).

Overall, our findings on all three empirical studies indicate that the Dynamic Synthetic Control (DSC) method offers significant advantages when compared to the standard Synthetic Control (SC) approach. In particular, DSC achieves closer approximations to the true trajectories of non-treated units, as evidenced by the observed reduction in Mean Squared Error (MSE). As a result, the DSC method provides a more accurate approximation of the true treatment effects. It is a more efficient estimator of the treatment effects and mitigates the risk of biased estimates.

5. Conclusion

In this study, we tackle the issue of varying reaction speeds across different units when estimating causal effects of treatments. This issue arises from the fact that units—such as states, cities, or people—may adjust and react at different speeds due to a multitude of factors, including legal constraints, institutional differences, and the nature of the treatment itself. Ignoring speed variations can bias estimates and conclusions, weakening the validity of treatment effect analyses.

We introduce the Dynamic Synthetic Control (DSC) method, which extends the synthetic control approach by integrating a Dynamic Time Warping (DTW) algorithm to adjust for speed differences. By doing so, the DSC method enables researchers to construct counterfactuals that more accurately represent the hypothetical outcomes for treated units without the treatment, with

is due to the warping adjustments made to account for varying speeds. Some of the time series are warped to a shorter length due to their slower speeds relative to the target time series (See Appendix 9 for more detail).

improved precision and efficiency of treatment effect estimates compared to the standard synthetic control method.

Through Monte Carlo simulations and real-world datasets, we show that DSC outperforms standard synthetic control in treatment effect estimation. It reduces uncertainty and boosts test power, minimizing the risk of false conclusions. These results highlight the value of our approach in addressing the speed problem and improving the accuracy and precision of treatment effect estimates.

Future research will be needed to explore the applications of DSC in various settings, such as assessing the impact of policies, interventions, or shocks across multiple dimensions and over different time horizons. Additional research could also assess DSC's sensitivity to unobservable variables and endogenous regressors.

References

Abadie, Alberto. 2021. Using synthetic controls: feasibility, data requirements, and methodological aspects. *Journal of Economic Literature* 59 (2): 391–425.

Abadie, Alberto, and Matias D Cattaneo. 2021. *Introduction to the special section on synthetic control methods,* 536.

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.

———. 2015. Comparative politics and the synthetic control method. *American Journal of Political Science* 59 (2): 495–510.

Abadie, Alberto, and Javier Gardeazabal. 2003. The economic costs of conflict: a case study of the basque country. *American Economic Review* 93 (1): 113–132.

Alesina, Alberto, Gerald D Cohen, and Nouriel Roubini. 1992. Macroeconomic policy and elections in oecd democracies. *Economics & Politics* 4 (1): 1–30.

Athey, Susan, and Guido W Imbens. 2017. The state of applied econometrics: causality and policy evaluation. *Journal of Economic Perspectives* 31 (2): 3–32.

Barro, Robert J, Xavier Sala-i-Martin, Olivier Jean Blanchard, and Robert E Hall. 1991. Convergence across states and regions. *Brookings papers on economic activity,* 107–182.

Ben-Michael, Eli, Avi Feller, and Jesse Rothstein. 2022. Synthetic controls with staggered adoption. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 84 (2): 351–381.

Brady, Henry E, and John E McNulty. 2011. Turning out to vote: the costs of finding and getting to the polling place. *American Political Science Review* 105 (1): 115–134.

Canarella, Giorgio, Stephen Pollard, et al. 2004. Parameter heterogeneity in the neoclassical growth model: a quantile regression approach. *Journal of Economic Development* 29:1–32.

Card, David, and Alan B Krueger. 2000. Minimum wages and employment: a case study of the fast-food industry in new jersey and pennsylvania: reply. *American Economic Review* 90 (5): 1397–1420.

Chattopadhyay, Raghabendra, and Esther Duflo. 2004. Women as policy makers: evidence from a randomized policy experiment in india. *Econometrica* 72 (5): 1409–1443.

Di Tella, Rafael, and Ernesto Schargrodsky. 2004. Do police reduce crime? estimates using the allocation of police forces after a terrorist attack. *American Economic Review* 94 (1): 115–133.

Ferman, Bruno, Cristine Pinto, and Vitor Possebom. 2020. Cherry picking with synthetic controls. *Journal of Policy Analysis and Management* 39 (2): 510–532.

Fleming, Michael J, and Eli M Remolona. 1999. Price formation and liquidity in the us treasury market: the response to public information. *The journal of Finance* 54 (5): 1901–1915.

Goodman-Bacon, Andrew. 2021. Difference-in-differences with variation in treatment timing. *Journal of Econometrics* 225 (2): 254–277.

Greene, William H. 2003. *Econometric analysis.* Pearson Education India.

Öztekin, Özde, and Mark J Flannery. 2012. Institutional determinants of capital structure adjustment speeds. *Journal of Financial Economics* 103 (1): 88–112.

Rappaport, Jordan. 2000. Is the speed of convergence constant? *FRB of Kansas City Research Working Paper No. 00-10.*

Reinhart, Carmen M, and Kenneth S Rogoff. 2014. Recovery from financial crises: evidence from 100 episodes. *American Economic Review* 104 (5): 50–55.

Sakoe, Hiroaki, and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing* 26 (1): 43–49.

Vintsyuk, Taras K. 1968. Speech discrimination by dynamic programming. *Cybernetics* 4 (1): 52–57.

Supplementary Material

Appendix 1. The speed problem with lags and polynomials

In an attempt to improve the synthetic control's approximation to the treated unit $\mathbf{y}_1$, one might consider enriching the model by adding time lags and polynomial terms. Specifically, in addition to the original control units $\mathbf{y}_2$ and $\mathbf{y}_3$, we extend the set of potential controls to include their lags (up to lag 5) and polynomial terms (up to the 5th degree). These extensions aim to capture additional dynamics and non-linearities in the data.

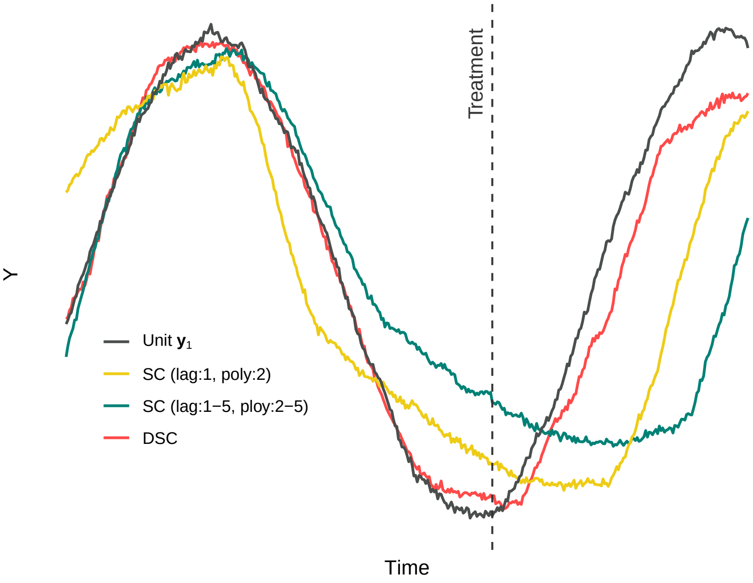

Figure A1. The Challenge of Varying Speeds in Treatment Effect Estimation. A researcher aims to quantify the impact of a treatment on unit $\mathbf{y}_1$. Unbeknownst to them, no treatment effect actually exists. When employing standard synthetic control methods that incorporate control units $\mathbf{y}_2$ (slow) and $\mathbf{y}_3$ (fast) along with their lags (1–5) and polynomial terms (2–5), the estimated post-treatment effect (represented by the blue curve) significantly diverges from the true outcome (indicated by the black curve). In contrast, the Dynamic Synthetic Control method produces an estimated synthetic control that more closely approximates the true trajectory.

As illustrated in Figure A1, however, the inclusion of these additional terms does not appreciably improve the performance of the synthetic control. The blue and yellow curves represent the synthetic control derived from expanded sets of potential controls. The blue curve, for example, solves the following:

$$W^* = \text{argmin}_{W^*} \left\| \mathbf{y}_1 - \left(\mathbf{Y}_{-1}\boldsymbol{\beta}^* + \sum_{l=1}^{5} L^l \mathbf{Y}_{-1}\boldsymbol{\gamma}^* + \sum_{k=2}^{5} \mathbf{Y}_{-1}^{k}\boldsymbol{\gamma}^* \right) \right\|_2,$$

where L^l is the lag operator.

However, both the blue and the yellow curves still deviate from the true trajectory (black curve) of unit $\mathbf{y}_1$. Consequently, one might still erroneously infer a treatment effect where none exists, highlighting the limitations of standard synthetic control methods in handling varying speeds among control units.

Appendix 2. Synthetic Controls: A refresher

Formally, suppose that we have data for $J + 1$ units.[14] For time $t \in (1, 2, \ldots, N)$, We observe a target unit $y_{1,t}$ that receives a treatment, and J units $y_{j,t}, j \in (2, 3, \ldots, J + 1)$ that are untreated and can therefore be used as donor units. Each unit is observed for N periods, and we assume that the treatment takes place at time T. For each unit, we might also observe a set of k predictors $z_{1,j}, z_{2,j} \ldots, z_{k,j}$, although for simplicity we do not include them in the model below.[15] Formally, a synthetic control can be represented by a $J \times 1$ vector of weights, $\mathbf{w} = (w_2, w_3, \ldots, w_{J+1})'$, and the synthetic control estimator of $y^N_{1,t}$ (the potential response in the absence of intervention) is then:

$$\widehat{y}^N_{1,t} = \sum_{j=2}^{J+1} w_j y_{j,t}$$

In other words, synthetic controls are weighted averages of the units in the pool of donors. The estimated effect of intervention is then:

$$\widehat{\tau}_t = y_{1,t} - \widehat{y}^N_{1,t}$$

The key question is how to choose the weights w_j. There are many possibilities, ranging from assigning equal weights to all control units to using a population-weighted unit. A sensible approach is to choose $w_2, w_3, \ldots, w_{J+1}$ such that the resulting synthetic control $\widehat{y}^N_{1,t}$ is as close as possible to the pre-intervention time series for the treated unit as possible (Abadie and Gardeazabal 2003; Abadie, Diamond, and Hainmueller 2010). By minimizing the distance between the trajectory of the treated unit and the combined untreated units, we build a synthetic control that is as close as possible to the (unobservable) counterfactual.

[14] This section largely follows Abadie (2021).

[15] The predictors can be added after warping for time-constant predictors, or the same algorithm described below can be applied to these predictors for time-varying variables.

Appendix 3. Bias of the synthetic control estimate when the time series have different speeds

In this appendix, we demonstrate that the standard synthetic control estimator may produce biased results when different units respond to shocks at different speeds. Mathematical proofs and models are presented here to support these claims.

Suppose that the target time series $\mathbf{y}_1$ and the donors $\mathbf{y}_j, j \in (2, 3, ..., J + 1)$ are of length N. Let T denote the treatment time and let the treatment effect be zero. To obtain weights $\mathbf{w}$, the synthetic control method essentially estimates the following model:

$$y_{1,t} = \sum_{j=2}^{J} (w_j y_{j,t}) + \epsilon_t, t < T$$

if the usual assumptions of the error term hold, then the least square estimate of the weights $\hat{\mathbf{w}}$ is unbiased and efficient, and can produce a synthetic control that is the closest to the target time series $\mathbf{y}_1$.

However, when the time series have different speeds, i.e. each time series has its own delays to responding to the common exogenous shocks $\mathbf{z}$, and the delays vary over time, then the usual assumptions of the error term in model (4) are violated and it introduces bias to $\hat{\mathbf{w}}$.

Taking the different speeds into account, model (11) becomes:

$$\begin{aligned} y_{1,t} &= \sum_{j=2}^{J} [w_j (y_{j,t} + \beta_{1,j,t} y_{j,t-1} + \beta_{2,j,t} y_{j,t-2} + \ldots \beta_{t-1,j,t} y_{j,1})] + \epsilon_t \\ &= (\mathbf{Y}_{-1,t} + \mathbf{Y}_{-1,t-1} \boldsymbol{\beta}_{1,t} + \ldots + \mathbf{Y}_{-1,1} \boldsymbol{\beta}_{t-1,t}) \mathbf{w} + \epsilon_t \end{aligned}$$

where $\mathbf{Y}_{-1,t}$ is a vector of donor units at time t, $\mathbf{Y}_{-1,t} = (y_{2,t}, y_{3,t}, ..., y_{J,t})$, and $\mathbf{Y}_{-1,t-1}$, $\mathbf{Y}_{-1,t-2}$, ... , $\mathbf{Y}_{-1,1}$ are all possible lags at time t. $\mathbf{w}$ is the time-independent coefficient. $\boldsymbol{\beta}_{l,t}$ are time-dependent coefficients of the lags. $\epsilon \sim N(0, \sigma^2 I)$ is the error term.

Model (13) allows a varying dependency of the target variable $\mathbf{y}_1$ on lags of the donor variables $\mathbf{Y}_{-1}$. This varying dependency captures the different speeds that $\mathbf{y}_1$ and $\mathbf{Y}_{-1}$ respond to the common exogenous shocks $\mathbf{z}$.

If the time series have different speeds and (13) is the true model, the original synthetic control method would obtain biased estimates of the treatment effect as the lag terms are omitted.

Pre-treatment Period

In traditional synthetic control analysis, the model of interest is,

$$y_{1,t} = \mathbf{Y}_{-1,t}\mathbf{w}_{sc} + \eta_t$$

the new error term η_t is

$$\eta_t = \mathbf{Y}_{-1,t-1}\boldsymbol{\beta}_{1,t}\mathbf{w} + \ldots + \mathbf{Y}_{-1,1}\boldsymbol{\beta}_{t-1,t}\mathbf{w} + \epsilon_t,$$

which is the summation of the original error term ϵ_t and the lag terms $\mathbf{Y}_{-1,t-1}\boldsymbol{\beta}_{1,t}\mathbf{w}+\ldots+\mathbf{Y}_{-1,1}\boldsymbol{\beta}_{t-1,t}\mathbf{w}$.

According to Greene 2003, the expectation of $\widehat{\mathbf{w}}_{sc}$ is

$$E[\widehat{\mathbf{w}}_{sc}] = \mathbf{w} + \mathbb{G}\boldsymbol{\beta}_L\mathbf{w}$$

where $\mathbb{G} = (\mathbf{Y}_{-1}'\mathbf{Y}_{-1})^{-1}\mathbf{Y}_{-1}'\mathbf{Y}_{-1,L}$. $\mathbf{Y}_{-1,L}$ is a matrix of lags of $\mathbf{Y}_{-1}$, and $\boldsymbol{\beta}_L$ is matrix of corresponding coefficients.

The expectation of residual

$$\begin{aligned} E[\widehat{\eta}] &= E[\mathbf{Y}_{-1,L}\boldsymbol{\beta}_L + \boldsymbol{\epsilon}] \\ &= E[\mathbf{Y}_{-1,L}\boldsymbol{\beta}_L] + 0 \end{aligned}$$

depends on the existence of lag effects. If lag effects exist, i.e. $E[\mathbf{Y}_{-1,L}\boldsymbol{\beta}_L] \neq 0$, $\widehat{\eta}$ is biased.

The variance of residual

$$\begin{aligned} \text{Var}(\widehat{\eta}) &= \text{Var}(\mathbf{Y}_{-1,L}\boldsymbol{\beta}_L + \boldsymbol{\epsilon}) \\ &= \text{Var}(\mathbf{Y}_{-1,L}\boldsymbol{\beta}_L) + \text{Var}(\boldsymbol{\epsilon}) + 0 \\ &= \text{Var}(\mathbf{Y}_{-1,L}\boldsymbol{\beta}_L) + \sigma^2 I \\ &\geq \sigma^2 I \end{aligned}$$

is at least $\sigma^2 I$. If the lag effects exist, i.e. $E[\mathbf{Y}_{-1,L}\boldsymbol{\beta}_L] \neq 0$, $\text{Var}(\widehat{\eta}) > \sigma^2 I$.

On the contrary, in dynamic synthetic control, the model of interest is:

$$y_t = \mathbf{Y}^*_{-1,t}\mathbf{w}_{dsc} + \epsilon_t$$

where

$$\mathbf{Y}^*_{-1,t} = \mathbf{Y}_{-1,t} + \mathbf{Y}_{-1,t-1}\boldsymbol{\beta}_{1,t} + \ldots + \mathbf{Y}_{-1,1}\boldsymbol{\beta}_{t-1,t}. \tag{A1}$$

The expectation of $\widehat{\mathbf{w}}_{dsc}$ is

$$E[\widehat{\mathbf{w}}_{dsc}] = \mathbf{w}$$

Note the error term is the same as in model (2). so the expectation of residual is just

$$E[\widehat{\epsilon}] = 0$$

and the variance of residual

$$\mathrm{Var}(\widehat{\epsilon}) = \sigma^2 I$$

Therefore, if time series have different speeds, i.e. effects of lags of $\mathbf{Y}_{-1}$ are not zero, in the pre-treatment period, traditional synthetic control method would fail to obtain a synthetic control that closely resembles $\mathbf{y}_1$.

Post-treatment Period

In post-treatment period, both synthetic control methods estimates the treatment effect using post-treatment $\mathbf{y}_1, \mathbf{Y}_{-1}$ and pre-treatment $\widehat{\mathbf{w}}_{sc}, \widehat{\mathbf{w}}_{dsc}$. For cleaner notification, let A denote the pre-treatment period and B denote the post-treatment period.

The estimated treatment effect from traditional synthetic control is:

$$\widehat{\boldsymbol{\tau}}_{sc} = \mathbf{y}_1^B - \mathbf{Y}_{-1}^B\widehat{\mathbf{w}}_{sc}^A$$

The expectation of $\hat{\boldsymbol{\tau}}_{sc}$ is

$$\begin{aligned} E[\hat{\boldsymbol{\tau}}_{sc}] &= E[\mathbf{y}_1^B - \mathbf{Y}_{-1}^B \hat{\mathbf{w}}_{sc}^A] \\ &= E[(\mathbf{Y}_{-1}^B \mathbf{w} + \mathbf{Y}_{-1,L}^B \boldsymbol{\beta}_L^B) - \mathbf{Y}_{-1}^B \hat{\mathbf{w}}_{sc}^A] \\ &= E[\mathbf{Y}_{-1,L}^B \boldsymbol{\beta}_L^B + \mathbf{Y}_{-1}^B (\mathbf{w} - \mathbf{w} - \mathbb{G}^A \boldsymbol{\beta}_L^A \mathbf{w})] \\ &= E[\mathbf{Y}_{-1,L}^B \boldsymbol{\beta}_L^B - \mathbf{Y}_{-1}^B \mathbb{G}^A \boldsymbol{\beta}_L^A \mathbf{w}] \end{aligned}$$

Whether the expectation of $\hat{\boldsymbol{\tau}}_{sc}$ equals to zero depends on $\mathbf{Y}_{-1}$, $\mathbf{Y}_{-1,L}$, $\boldsymbol{\beta}_L$, and $\mathbf{w}$. There are two sources of bias. One is omitted lag effects, $\mathbf{Y}_{-1,L}^B \boldsymbol{\beta}_L^B$. The other is biased estimate of weights, $-\mathbf{Y}_{-1}^B \mathbb{G}^A \boldsymbol{\beta}_L^A \mathbf{w}$. The two sources of bias are both zeros only when the lag effects are zero, i.e. $E[\mathbf{Y}_{-1,L} \boldsymbol{\beta}_L] = 0$. Otherwise, as the two sources of bias are only marginally correlated ($\mathbf{Y}_{-1}$ and $\mathbf{Y}_{-1,L}$ have some common terms), it is very rare that the two parts have different signs and in the meantime add up to zero, which means the estimated treatment effect from traditional synthetic control is highly likely biased.

The variance of $\hat{\boldsymbol{\tau}}_{sc}$

$$\begin{aligned} \mathrm{Var}[\hat{\boldsymbol{\tau}}_{sc}] &= \mathrm{Var}(\mathbf{y}_1^B - \mathbf{Y}_{-1}^B \hat{\mathbf{w}}_{sc}^A) \\ &= \mathrm{Var}[(\mathbf{Y}_{-1}^B \mathbf{w} + \mathbf{Y}_{-1,L}^B \boldsymbol{\beta}_L^B + \boldsymbol{\epsilon}) - \mathbf{Y}_{-1}^B \hat{\mathbf{w}}_{sc}^A] \\ &= \mathrm{Var}[\mathbf{Y}_{-1,L}^B \boldsymbol{\beta}_L^B - \mathbf{Y}_{-1}^B \mathbb{G}^A \boldsymbol{\beta}_L^A \mathbf{w} + \boldsymbol{\epsilon}] \\ &= \mathrm{Var}[\mathbf{Y}_{-1,L}^B \boldsymbol{\beta}_L^B - \mathbf{Y}_{-1}^B \mathbb{G}^A \boldsymbol{\beta}_L^A \mathbf{w}] + \sigma^2 I \end{aligned}$$

is larger or equal to $\sigma^2 I$. The equal sign holds only when $\hat{\boldsymbol{\tau}}_{sc}$ is not biased, i.e. (25) equals zero.

Instead, the estimated treatment effect from dynamic synthetic control is

$$\hat{\boldsymbol{\tau}}_{dsc} = \mathbf{y}_1^B - \mathbf{Y}_{-1}^{*B} \hat{\mathbf{w}}_{dsc}^A$$

The expectation of $\widehat{\boldsymbol{\tau}}_{dsc}$ is

$$\begin{aligned} E[\widehat{\boldsymbol{\tau}}_{dsc}] &= E[\mathbf{y}_1^B - \mathbf{Y}_{-1}^{*B}\widehat{\mathbf{w}}_{dsc}^A] \\ &= E[\mathbf{y}_1^B - \mathbf{Y}_{-1}^{*B}\mathbf{w}] \\ &= E[\boldsymbol{\epsilon}^B] \\ &= 0 \end{aligned}$$

The variance of $\widehat{\boldsymbol{\tau}}_{dsc}$ is

$$\begin{aligned} \mathrm{Var}[\widehat{\boldsymbol{\tau}}_{dsc}] &= \mathrm{Var}(\mathbf{y}_1^B - \mathbf{Y}_{-1}^{*B}\widehat{\mathbf{w}}_{dsc}^A) \\ &= \mathrm{Var}[(\mathbf{Y}_{-1}^B\mathbf{w} + \mathbf{Y}_{-1,L}^B\boldsymbol{\beta}_L^B + \boldsymbol{\epsilon}) - \mathbf{Y}_{-1}^B\widehat{\mathbf{w}}_{sc}^A] \\ &= \mathrm{Var}(\mathbf{y}_1^B - \mathbf{Y}_{-1}^{*B}\mathbf{w}) \\ &= \mathrm{Var}(\boldsymbol{\epsilon}^B) \\ &= \sigma^2 I \end{aligned}$$

Therefore, when time series have different speeds, dynamic synthetic control method can produce unbiased and efficient estimate of the treatment effect while the traditional synthetic control method would most likely produce biased estimates and inflated variances.

Appendix 4. The Forward Lag Coefficients

In model 1, $y_{1,t}$ can be influenced not only by past values of $y_{j,t}$, but also by its future values. This does not suggest that the future impacts the past. Instead, suppose that both $\mathbf{y}_j$ and $\mathbf{y}_1$ are functions of a latent variable $\mathbf{z}$—a variable representing true underlying shocks—while the observed time series for units $\mathbf{y}_j$ and $\mathbf{y}_1$ are the manifestations of the shocks. In this situation, the observed time series are always slowed-down versions of $\mathbf{z}$, but at different rates. For example, the time series for $\mathbf{y}_1$ may reflect changes in $\mathbf{z}$ immediately, whereas there may be a delay for $\mathbf{y}_j$. As a result, we would observe that $y_{1,t}$ is a function of the future of $y_{j,t}$, simply because $\mathbf{y}_1$ reacts faster to changes in $\mathbf{z}$ than $\mathbf{y}_j$ does. This phenomenon, for example, occurs in the US financial markets, where the reactions of the bond and stock markets to sudden changes in the Federal Reserve's monetary policies often occur at different speeds (Fleming and Remolona 1999). After the announcement of an interest rate hike, the bond market reacts to the news instantaneously, while the stock market requires active trading to adjust prices. In such cases, we observe a strong correlation between current bond prices and stock prices shortly thereafter. This means that, in order to correct for the relative speeds of $\mathbf{y}_j$ and $\mathbf{y}_1$, we need not only consider past lags of $\mathbf{y}_j$, but also forward lags. The forward lags have non-zero coefficients when $\mathbf{y}_1$ is faster than $\mathbf{y}_j$. However, note that since $\mathbf{y}_1$ and $\mathbf{y}_j$ are always slower than the latent common exogenous shocks $\mathbf{z}$, the forward lags of $\mathbf{y}_j$ are associated with exogenous shocks that have already occurred, not future events. We do not suggest that the future affects the past, but rather simply that $\mathbf{y}_j$ and $\mathbf{y}_1$ react at different speeds to $\mathbf{z}$, and hence that $\mathbf{y}_1$ may *appear* to be a function of future $\mathbf{y}_j$. This implies that there are really a total of JNN potential coefficients to estimate ((J donor time series) × (N time periods) × ($N/2$ backward lags + $N/2$ forward lags)).

Appendix 5. The Dynamic Synthetic Control Algorithm

Algorithm A1 Dynamic Synthetic Control (DSC)

```
Input: $\mathrm{y}_1 = \{y_{1,1}, y_{1,2}, ..., y_{1,N}\}$: time series with treatment.
       $\mathrm{y}_j = \{y_{j,1}, y_{j,2}, ..., y_{j,N}\}$: J time series without treatment.
       $T$: treatment time.
       $C_j$: cutoff time of $\mathrm{y}_j$.
       $k$: size of the sliding window.
       $m$: margin of the sliding window.
       $n_Q$: increment for sliding the target window.
       $n_R$: increment for sliding the reference window.
       $\theta$: threshold of distance in window matching.
Output: synthetic control $\mathrm{y}_1^N$.
Process:
$j \leftarrow 1$
for $j = 2$ to $J + 1$ do
        Step 1: Match $\mathrm{y}_{1,1:T}$ and $\mathrm{y}_j$;
            DTW match $\mathrm{y}_{1,1:T}$ and $\mathrm{y}_j$ with $open.end = TRUE$,
            save $\mathrm{y}_{j,1:C_j}$ that matches $\mathrm{y}_{1,1:T}$ as $\mathrm{y}_{j,pre}$,
            save $\mathrm{y}_{j,C_j:N}$ as $\mathrm{y}_{j,post}$, $\mathrm{y}_{1,1:T}$ as $\mathrm{y}_{1,pre}$, and $\mathrm{y}_{1,T:N}$ as $\mathrm{y}_{1,post}$
            save warping path $\mathrm{y}_{j,pre} \rightarrow \mathrm{y}_{1,pre}$ as $\mathbf{P}_{j,pre}$,
        Step 2: Match $\mathrm{y}_{j,post}$ and $\mathrm{y}_{j,pre}$ (double sliding window);
            $u \leftarrow 1; \mathbb{P}_i \leftarrow NULL$
            while $u \leq length(\mathrm{y}_{j,post}) - k$ do
                locate target window $Q_u = \mathrm{y}_{j,post}[u : (u + k)]$,
                $i \leftarrow 1; costs \leftarrow NULL$
                while $i \leq length(\mathrm{y}_{j,pre}) - k - m$ do
                    locate reference window $R_i = \mathrm{y}_{j,pre}[i : (i + k + m)]$,
                    DTW match $Q_u$ and $R_i$ with $open.end = TRUE$,
                    save DTW distance to $costs$,
                    $i \leftarrow i + n_R$.
                end while
                if $min(costs) \leq \theta$ then
                    find $i^*$ that minimizes $costs$,
                    locate reference window $R^* = \mathrm{y}_{j,pre}[i^* : (i^* + k + m)]$,
                    subset $\mathbf{P}_{j,pre}$ and obtain warping path $\mathbf{P}_{j,R^*}$,
                    save warping path $Q_u \rightarrow R^*$ as $\mathbf{P}_{j,Q_u \rightarrow R^*}$,
                    obtain warping path $\mathbf{P}_{j,Q_u} = \mathbf{P}_{j,Q_u \rightarrow R^*}(\mathbf{P}_{j,R^*})$,
                    store $\mathbf{P}_{j,Q_u}$ in list $\mathbb{P}_j$,
                end if
                $u \leftarrow u + n_Q$.
            end while
            merge all $\mathbf{P}_{j,Q_u}$ in $\mathbb{P}_j$, obtain $\mathbf{P}_{j,post}$.
        Step 3: Warp $\mathrm{y}_j$ according to $\mathrm{y}_1$
            $\mathrm{y}_j^w = [\mathbf{P}_{j,pre}(\mathrm{y}_{j,pre}), \mathbf{P}_{j,post}(\mathrm{y}_{j,post})]$.
end for
Step 4: use $\mathrm{y}_j^w, j \in (2, \ldots, J + 1)$ to construct synthetic control $\mathrm{y}_1^N$.
return $\mathrm{y}_1^N$
```

Appendix 6. Deriving post-treatment warping paths from pre-treatment warping paths

To derive the warping path $\mathbf{P}_{j,post}$ from $\mathbf{P}_{j,pre}$, we first match the short-term patterns in $\mathbf{y}_{j,post}$ to the patterns in $\mathbf{y}_{j,pre}$, and for each matched pattern in $\mathbf{y}_{j,pre}$, we obtain a sub-warping path matrix by subsetting $\mathbf{P}_{j,pre}$; then we adjust the sub-warping path matrix according to the DTW matching from the pattern in $\mathbf{y}_{j,post}$ to the pattern in $\mathbf{y}_{j,pre}$; lastly, we combine the adjusted sub-warping path and obtain $\mathbf{P}_{j,post}$.

To find the similar patterns in $\mathbf{y}_{j,post}$ and $\mathbf{y}_{j,pre}$, we use a double-sliding windows approach as shown in the second part of Figure 2. The first sliding window is in $\mathbf{y}_{j,post}$ and the second in $\mathbf{y}_{j,pre}$, Specifically, for any target window Q_u in the post-treatment time series $\mathbf{y}_{j,post}$, the method finds an optimal reference window R^* in the pre-treatment time series $\mathbf{y}_{j,pre}$ that minimizes the DTW distance between Q_u and R_i. Pattern R^* is therefore the closest match to pattern Q_u:

$$\begin{aligned} R^* &= \text{argmin}_{R_i}[DTW(Q_u, R_i)], \\ Q_u &= \mathbf{y}_{j,u:(u+k)}, u \in [C, N-k], k \in \mathbb{Z}^+ \\ R_i &= \mathbf{y}_{j,i:(i+k+m)}, i \in [1, C-k-m], m \in \mathbb{Z}^0 \end{aligned}$$

where k is the default size of the sliding windows, and m is an extra right boundary of the reference window R_i for the open-end DTW, i.e. allowing the the target window Q_u to freely match to a pattern that is within R_i, which enables the DSC algorithm to consider patterns in different lengths. m is chosen by the user to ensure it is large enough to have the matching achieved within R_i, while as small as possible to reduce the computational burden.

We recommend using slope-constrained step patterns such as *symmetricP2* or *asymmetricP2* in DTW to avoid extreme warping paths. In addition, all windows are normalized before DTW for better matching results.

If no similar pattern can be found in $\mathbf{y}_{j,pre}$, i.e. $min(costs)$ is too large, R^* can not provide enough information to help build the warping path $\mathbf{P}_{j,post}$. Instead, including poor matches would introduce noise to the warping path. To avoid this problem, we use a threshold θ to filter out the poor matches.

For each pattern Q_u that has a close match R^* whose DTW distance does not exceed the threshold θ, the warping path $Q_u \rightarrow R^*$ is stored in a $k \times (k+m)$ matrix $\mathbf{P}_{j,Q_u}$. Although Q_u and R^* are close matches, they are not identical. We can not directly use warping path $\mathbf{P}_{j,R^*}$ (i.e. sub-matrix of $\mathbf{P}_{j,pre}$

that warps R^* towards $\mathbf{y}_{1,pre}$) to warp Q_u. We need to adjust $\mathbf{P}_{j,R^*}$ and take the difference between Q_u and R^* into account.

To help transform $\mathbf{P}_{j,R^*}$ using $\mathbf{P}_{j,Q_u}$, we can transform the warping path matrix $\mathbf{P}_{j,pre}$ into a Speed Profile (SP):

$$\phi_{j,pre} = \{\mathbf{P}_{j,pre}[diag(\mathbf{I}_{1\times C}\mathbf{P}_{j,pre})^{-1}]\}\mathbf{I}_{T\times 1}$$

where $\mathbf{I}_{1\times C}$ is a row of ones of length C, and similarly, $\mathbf{I}_{T\times 1}$ is a column of ones of length T.

A speed profile can also be seen as a one-dimensional version of the warping path matrix. It generates the same warped time series as the warping path matrix does. The difference between a speed profile and a warping path matrix is that the speed profile only supports single-direction warping, while the latter supports both directions. Since we are not interested in warping $\mathbf{y}_{1,pre} \to \mathbf{y}_{j,pre}$, in our case, an one-dimensional speed profile works as well as a warping path matrix. And most importantly, it greatly simplifies the process of deriving the warping path $\mathbf{P}_{j,post}$, as we can use a sub-speed profile to warp the corresponding window of time series and ignore the remaining speed profile without losing any information. In our method, a reference window $R_i = \mathbf{y}_{j,i:(i+k+m)}$, $i \in [1, C-m-k]$ can be warped towards $\mathbf{y}_{1,pre}$ using the sub-speed profile $\phi_{R_i} = \phi_{j,pre}[i : (i + k + m)]$.

The DSC algorithm then uses the warping path $\mathbf{P}_{j,Q_u} : Q_u \to R^*$, to transform the sub-speed profile ϕ_{j,R^*} and obtains a new speed profile ϕ_{j,Q_u} for $Q_u \to \mathbf{y}_{1,post}$. The speed profile ϕ_{j,Q_u} on one hand preserves the speed relationship between $\mathbf{y}_{j,pre}$ and $\mathbf{y}_{1,pre}$, on the other hand, makes adjustments and makes sure the difference between Q_u and R^* is taken into account. Specifically, without losing generality, in matching target window Q_u and its closest reference window R^*, if one data point $Q_u[s]$ matches one or more data points $R^*[p : q]$, we define the weight of $Q_u[s]$ in speed profile is the average of the weights of $R^*[p : q]$. Formally,

$$\phi_{j,Q_u}[s] = \frac{\sum_{i\in[p,q]} \phi_{j,R^*}[i]}{q-p+1}, s \in [1,k], 1 \le p \le q \le (k+m)$$

The transformed speed profile ϕ_{j,Q_u} can be obtained through the following process:

$$\phi_{j,Q_u} = \{[diag(\mathbf{P}_{j,Q_u}\mathbf{I}_{(k+m)\times 1})^{-1}]\mathbf{P}_{j,Q_u}\}\phi_{j,R^*}$$

The resulting speed profiles ϕ_{j,Q_u}, $u \in [C, N - k]$ are stacked in a $(N - C - k + 2) \times (N - C + 1)$ matrix $\boldsymbol{\phi}_j$:

$$\boldsymbol{\phi}_j = [\phi_{j,\xi,\upsilon}], \xi \in [1, N - C - k + 2], \upsilon \in [1, N - C + 1]$$
$$\phi_{j,\xi,\upsilon} = \phi_{j,Q_u}[\upsilon - \xi + 1]$$

And the estimate of the speed profile $\phi_{j,post}$ is the column mean of the stacked weight matrix $\boldsymbol{\phi}_j$:

$$\phi_{j,post} = \mathbf{I}_{1\times(N-C-k+2)}\boldsymbol{\phi}_j$$

Please see Appendix 7 for the method to warp time series using a speed profile.

Appendix 7. Warp time series using speed profile

As discussed in Appendix 6, the speed profile ϕ is an one-dimensional version of a warping path matrix $\mathbf{P}$:

$$\phi = \{\mathbf{P}[diag(\mathbf{I}_{1\times nrow(\mathbf{P})}\mathbf{P})^{-1}]\}\mathbf{I}_{ncol(\mathbf{P})\times 1}$$

It is used in the DSC algorithm to simplify the double-sliding window approach and can also be used to warp time series.

Given a speed profile ϕ, we can warp the corresponding time series $\mathbf{y}$ through the following transformation:

$$\mathbf{y}^{w} = \mathbf{y}(\phi) = \mathbf{H}\mathbf{y}$$

where $\mathbf{H}$ is a transformation matrix determined by the speed profile ϕ. The process for obtaining $\mathbf{H}$ from ϕ is shown in Algorithm A2.

Algorithm A2 Obtain transformation matrix H from speed profile ϕ

```
Input: $\phi = \{\phi_1, \phi_2, \ldots, \phi_N\}$: speed profile for $\mathbf{y}$.
       $N$: length of $\phi$.
Output: $H$: transformation matrix used to warp $\mathbf{y}$.
Process:
i ← 1; j ← 1; H ← NULL; r ← φ[1]; c ← 1
while $i \le N$ do
    $H[j, i] = min(r, c)$
    if $r > c$ then
        $j = j + 1$
        $r = r - c$
        $c = 1$
    end if
    if $r < c$ then
        $i = i + 1$
        $c = c - r$
        $r = \phi[i]$
    end if
    if $r == c$ then
        $i = i + 1$
        $j = j + 1$
        $c = 1$
        $r = \phi[i]$
    end if
end while
return $H$
```

For example, a speed profile $\phi = (1.5, 1, 0.5)$ gives a transformation matrix **H**:

$$\phi = \begin{bmatrix} 1.5 & 1 & 0.5 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \end{bmatrix} = H$$

Notice that the transformation matrix is just a matrix of lag coefficients $\beta_{l,t}$ in equation 1:

$$H = \begin{bmatrix} \beta_{0,1} = 1 & \beta_{-1,1} = 0 & \beta_{-2,1} = 0 \\ \beta_{1,2} = 0.5 & \beta_{0,2} = 0.5 & \beta_{-1,2} = 0 \\ \beta_{2,3} = 0 & \beta_{1,3} = 0.5 & \beta_{0,3} = 0.5 \end{bmatrix}.$$

And process **Hy** is equivalent to combining the lag terms into a new time series in equation A1.

Appendix 8. Data Generation in Monte Carlo Study

Each dataset contains ten time series. They all follow a common Autoregressive integrated moving average (ARIMA) process but exhibit differing speeds. To simulate these varying speeds, each time series j is warped by a matrix $\mathbf{P}_j$, which is either random or a function of the time series direction (increasing or decreasing). More formally,

$$\begin{aligned} \mathbf{y}_1 &= \mathbf{P}_1(\mathbf{z}) + \boldsymbol{\tau} + \boldsymbol{\epsilon}_1, \\ \mathbf{y}_j &= \mathbf{P}_j(\mathbf{z}) + \boldsymbol{\epsilon}_j, \\ \mathbf{P}_j &= f[\psi \mathbf{s}_j + (1-\psi)\boldsymbol{\lambda}_j] \\ \boldsymbol{\lambda}_j &\sim N(\overline{\mathbf{s}_j}, var(\mathbf{s}_j)), \end{aligned} \tag{A2}$$

where $\mathbf{z}$ denotes a 100 observation-long ARIMA(1,1,0) process. This process is warped by a warping path $\mathbf{P}_j$, which varies across units. The warping path $\mathbf{P}_j$ is a function of variable $\mathbf{s}_j$, which takes on value δ_j (a random variable specific to the unit) if $\mathbf{z}$ is increasing, and δ_j^{-1} otherwise. I.e.:

$$s_{j,t} = \begin{cases} \delta_j & \text{if } \mathbf{z}'_t > 0 \\ \frac{1}{\delta_j} & \text{otherwise,} \end{cases} \qquad \delta_j \sim U(a,b)$$

The idea behind $\mathbf{s}_j$ is to capture the possibility that speed may vary as a function of the direction of the underlying series. Economic crashes (i.e. a decreasing series), for example, may unfold faster than recoveries (Reinhart and Rogoff 2014).

The other term in the warping path expression, $\psi \in (0,1)$, determines the extent to which this occurs. For instance, $\psi = 0$ implies that the warping path will be entirely governed by a random normal variable λ. In this case, the speed at each observation will randomly differ from the next, and $\mathbf{s}_j$ will play no role. Conversely, if $\psi = 1$, the speed profile will completely depend on the direction of the time series (although the speed is still specific to each series, as the random draw δ_j is unit-specific).

Finally, τ_t denotes the value of the true treatment effect at time t. For the treated unit,

$$\tau_t = \begin{cases} 0 & \text{if } t \leq T \\ \frac{t-T}{10}\kappa & \text{if } t \in [T, T+10) \\ \kappa & \text{if } t \geq T+10, \end{cases}$$

where κ refers to the magnitude of the treatment effect, and $T = 60$ here. The chosen form of τ_t represents a shock that unfolds over several periods, capturing the gradual impact often observed in real-world scenarios. This modeling choice, not central to our approach, is simply intended to enhance the representation of actual shocks, whose effects may take time to materialize. An illustrative sample dataset is displayed in Figure A2.

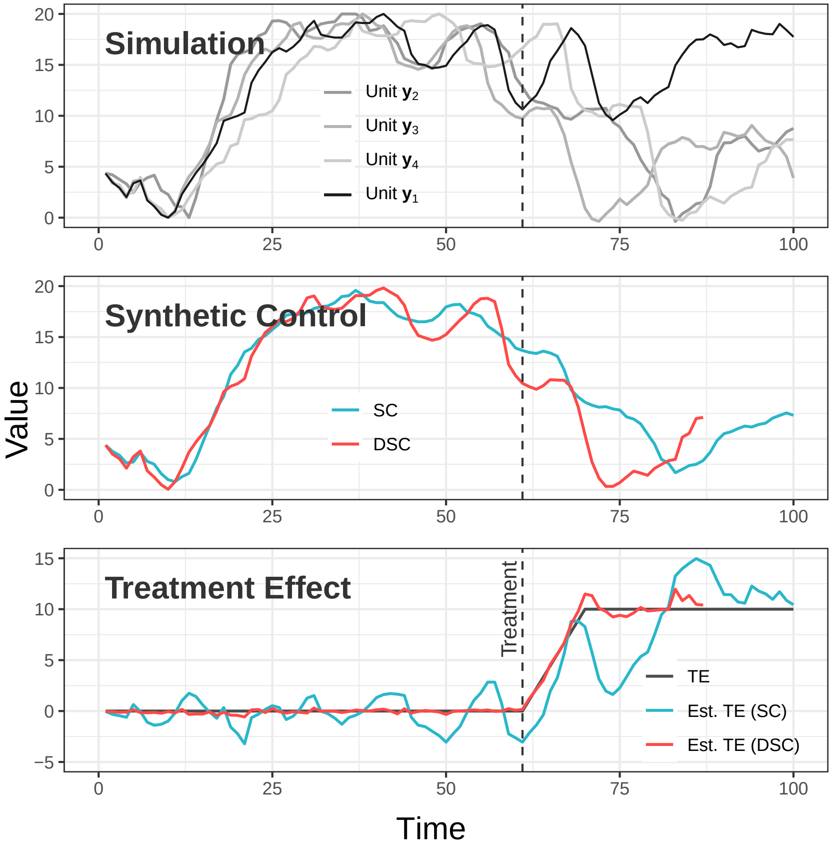

Figure A2. Illustration of an artificial dataset generated using the Model in equation A2. Only a subset of units is displayed for readability. Top: Units $\mathbf{y}_2$–$\mathbf{y}_4$ are used to construct counterfactuals approximating Unit $\mathbf{y}_1$ during the pre-treatment period ($t < 60$). The middle figure depicts synthetic controls: standard SC in blue and DSC in red. The lower figure shows that the estimated treatment effects from DSC are more accurate in capturing the true effect compared to those from the standard SC method.

Appendix 9. Length of warped time series

The length of the warped time series $\mathbf{y}_j^w$ often differs from the target time series $\mathbf{y}_1$ for two reasons: (i) the cutoff time C_j and the treatment time T are not necessarily the same; (ii) the warping of the post-treatment time series $\mathbf{y}_{j,post}$ is not based on a direct DTW matching between $\mathbf{y}_{j,post}$ and $\mathbf{y}_{1,post}$. While $\mathbf{y}_{j,pre}^w$ is guaranteed to be of length T, the length of $\mathbf{y}_{j,post}^w$ could be shorter or longer than $\mathbf{y}_{1,post}$ depending on whether C_j is smaller than T or the warping path $\mathbf{P}_{j,post}$ accelerates the average speed of $\mathbf{y}_{j,post}$, thus shortening the warped time series. The difference in length is normal, as the speeds of $\mathbf{y}_1$ and $\mathbf{y}_j$ can vary and we do not impose a strict end-rule that forces $y_{j,N}$ to match $y_{1,N}$. If $\mathbf{y}_{j,post}^w$ is shorter than $\mathbf{y}_{1,post}$, we lose some data points at the end of the time series but gain better speed alignment immediately after the treatment at T. This improved alignment is beneficial for comparative studies.

Changes in the length of the warped donor time series have significant implications for the implementation of the DSC method. In instances where the post-treatment time series are very short, if the DSC method yields a shorter synthetic control, there might be insufficient post-treatment data points to deduce a solid causal inference. Take, for instance, a scenario where the post-treatment segment consists of 12 periods. If, after applying the DSC method, the resulting synthetic control has only 8 periods, then conducting a causal effect study over 10 periods becomes unfeasible.

To avoid this issue, we recommend the adoption of slope-constrained step patterns like *symmetricP2* or *asymmetricP2* (Sakoe and Chiba 1978) in DSC so that the time series are not overly warped. Depending on the chosen step pattern, users can determine the minimum periods of the warped time series. For instance, after warping using the *symmetricP2* or *asymmetricP2* step patterns, the potential length of the warped time series lies within the $[\frac{2}{3}N, \frac{3}{2}N]$ range. This suggests that maintaining a buffer of $\frac{1}{3}N$ can guarantee that the warped time series retains adequate periods for subsequent analysis.

Appendix 10. R package

For the convenience of researchers and practitioners interested in employing the methods discussed in this paper, we have developed an accompanying R package. This package is publicly available at github.com/conflictlab/dsc. The package includes functions to implement all described methods, and it comes with comprehensive documentation to guide users through the installation and application process.

To install the package, one can execute the following R command:

```
devtools::install_github("conflictlab/dsc")
```

Additional documentation and examples for using the package can be found at https://github.com/conflictlab/dsc/blob/main/README.md.

Researchers are encouraged to refer to the package when utilizing the methods described in this paper, and a citation format for acknowledging the package is provided in the package documentation.