# Fit is all you need: Counterfactual Raster Generation for Spatial Intervention Evaluation

Daniel Lasso-Jaramillo[†‡]

This paper introduces CNN-SC, a Deep Learning methodology designed to estimate the effect of spatial interventions on raster outcomes while addressing potential violations of the Stable Unit Treatment Value Assumption (SUTVA), such as spillovers. Through the utilization of a Convolutional Neural Network (CNN), trained on pre-intervention data from control units, CNN-SC learns a latent representation of their post-intervention states. This learned representation enables the method to estimate the interference between a raster's pixel and its related pixels, treating potential outcomes as a linear combination of these related pixel outcomes. The optimal weights for this combination are learned through the CNN. Subsequently, this representation is applied to the pre-intervention data of treated units to generate a counterfactual raster that considers the joint estimation of potential outcomes. Notably, CNN-SC demonstrates the capability to estimate spillover effects without relying on prior assumptions regarding spillover propagation, thereby allowing for the identification of spillover dynamics post-spatial intervention. Through simulations, this paper validates that CNN-SC achieves a more precise estimation (deviation of 13%) of the real direct treatment effect compared to traditional causal inference techniques like Differences-in-Differences (deviation of 28% to 53%).

**Keywords**: Causal Machine Learning, Raster Generation, Remote Sensing, Spatial intervention, Convolutional Neural Networks.
**JEL Codes**: C01, C14, C33, C45

[†]Faculty of Economics, University of los Andes. Contact info: df.lasso@uniandes.edu.co
[‡]This article could not be possible without the help and comments from Manuel Fernandez and Ignacio Sarmiento. All errors and omissions are my own.

# 1 Introduction

How can we evaluate the impact of a spatial intervention on an outcome of interest using rasters? Literature related to spatial interventions approaches this issue on a similar basis. First, they extract tabular data from spatial sources such as maps or rasters, and then apply causal inference techniques such as Synthetic Control (SC), Regression Discontinuity (RD), or Differences in Differences (DiD) to identify causal effects across the interested outcomes (Fick, Nauman, Brungard, & Duniway, 2021; Gonzalez, 2021; Ruining, Shuai, & Lili, 2021). Yet, this two-stage approach encounters the following challenges. First, the Stable Unit Treatment Value Assumption (SUTVA) is frequently violated in spatial interventions. This occurs for two main reasons: the potential outcomes of a unit vary with the treatment assignment of other units (interference) (Imbens & Rubin, 2015). Also, due to spill-overs, estimates are biased if they do not consider the indirect effects on control units (Keele, 2015). This indirectly results in various problems regarding how to control for and correctly identify these spillovers (Pollmann, 2022). Another challenge is that by transforming the raster into tabular data, the spatial structure is omitted. Estimation is then constrained to a rigid unit level (i.e., pixel), precluding contextual estimation that may arise from understanding how features or entities as a whole are affected or spilled by treatment (i.e., rivers, forests, cities) (Sims & Alix-Garcia, 2017).

I propose a methodology called Convolutional Neural Networks Synthetic Control (CNN-SC), based on Synthetic Control Methods (SCM) and Deep Learning (DL), to generate a counterfactual raster that allows the evaluation of the effect of a spatial intervention on certain outcomes. I train a Deep Network (CNN) using control units' pre-intervention raster data to learn a latent representation of their post-intervention states. This representation is then applied to the treated unit's pre-intervention data to generate a counterfactual raster. The causal effect is estimated by contrasting the predicted raster with the observed one. Similarly, I argue that this method allows for flexible learning of interference across the raster's pixels and addresses the violation of SUTVA. Additionally, this method provides a flexible way of understanding spillovers and their propagation. For statistical inference, I propose a placebo test common in synthetic controls and randomization inference literature.

Results suggest that CNN-SC is able to flexibly estimate the interference across pixels within a raster through kernels and convolutions. This allows the creation of a proper counterfactual raster under this specific SUTVA violation. Moreover, preliminary results using Monte Carlo simulations suggest that this methodology

outperforms traditional DiD estimation with and without spillovers correction. For instance, the mean percentage deviation from the estimates of CNN-SC to the simulated treatment value was 13%, while the deviation for DiD was 53% and 28%, respectively. Additionally, the simulations show that CNN-SC can illustrate the spillovers propagation by analyzing the untreated pixels' treatment effect.

The proposed method contributes to the literature in several ways: i) it effectively deals with violations of SUTVA by leveraging Convolutional Neural Network (CNN) neurons, allowing for a data-driven characterization of interference across pixels. ii) it estimates the spillover effects for a given treatment without a priori assuming the geometry or spatial distribution of its propagation. iii) Also, it provides the investigator with a visual presentation of the spillover effects, enabling them to identify the propagation of the spillovers. iv) CNN-SC is the first article to leverage CNN into the synthetic control literature, providing insights into how to use convolutions and kernels to improve counterfactual prediction on raster data.

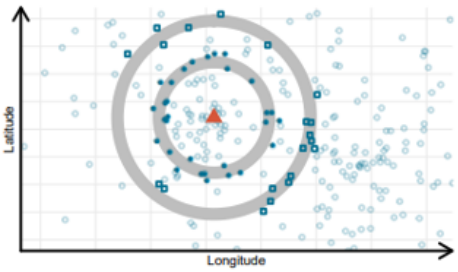
As previously explained, this paper addresses two main problems in spatial intervention literature. It seeks to solve some issues derived from SUTVA violations. Formally speaking, SUTVA has two restrictions. First, the potential outcomes of a unit do not vary with the treatment assignment of other units. Second, there are no hidden treatment variations. For example, there is no differential intensity from a unit to the treatment (Imbens & Rubin, 2015; Keele, 2015). In spatial interventions, it is unambiguous that the potential outcomes of a unit vary depending on the spatially related units' treatment status. On the other hand, if a unit is spatially related to multiple units, then the potential outcomes vary not only due to the dichotomous exposure or non-exposure but also because the potential outcomes vary depending on how many of these related units are treated (Pollmann, 2022). Both channels force the violation of SUTVA. I will argue that CNN-SC is able to surpass the first violation of SUTVA; however, it is unable to solve the second violation. This will generate difficulties for external validity and the interpretation of effects.

In spatial intervention literature, there's a focus on estimating or controlling spillover effects by approximating their propagation. Typically, it's assumed that spillovers affect units within a certain distance from the spatial treatment location. Often, an Euclidean distance measure is employed, resulting in the generation of circles or geometric features that determine whether a unit is exposed to the treated units or not. Figure 1 presents two examples of how Randomized Controlled Trials (RCT) and Differences-in-Differences (DiD) approaches handle spillovers by creating

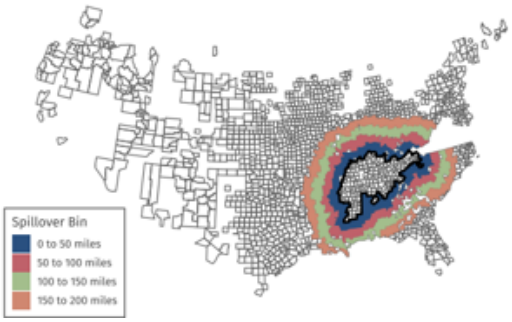
distance rings that separate the treated, control, and contaminated control groups. These distance rings are further used to estimate the spillover effects. However, this approach is very problematic. Consider, for example, that the spillover effects do not propagate through a Euclidean distance, but rather through other unobserved covariates such as rivers, roads, networks, etc. In this case, these methods are unable to accurately estimate the spillovers and may yield biased estimates. CNN-SC contributes to the literature by allowing for data-driven spillover estimation without making assumptions about how they propagate.

Figure 1: Examples of spillover estimation in the literature

(a) Spatial intervention on shops




(b) Tax related intervention




*Source*: panel a) taken from Pollmann (2022) identifies the spillover effects of a spatial intervention represented by a triangle. Panel b) taken from Butts (2023) presents an spatial intervation related to tax payment wich occurred on some states of the US.

On the other hand, CNN-SC method relates to the SCM as it fits in a category-bounding predictive technique used to create a synthetic counterfactual of a treated unit. Abadie and Gardeazabal (2003) introduced a method where the treated unit is represented as a linear weighted sum of untreated units, minimizing mean squared prediction error (MSPE). Other approaches have explored the possibility of using Machine Learning to improve the synthetic control. For example, by using elastic net to deal with multidimensional data (Doudchenko & Imbens, 2016; Brodersen, Koehler, Remy, Scott, & Gallusser, 2015). This paper differs from this literature as it is not focusing on creating a combination of untreated units, but to learn a latent representation of the effect of confounders on outcomes over time. Also, it is the firts paper that tries to create a synthetic control for a matrix.

It also relates to the Causal Machine Learning literature, which combines causal inference with novel machine learning techniques (Athey & Imbens, 2019; Hu-

ber, 2021). Especially relevant, Athey, Bayati, Doudchenko, Imbens, and Khosravi (2021) proposes an application of the Machine Learning method called Matrix Completion for causal panel data models. The idea is to impute the "missing" potential control outcomes of a time-series dependent matrix for the untreated unit. However, CNN-SC goes beyond this literature, as its main objective is not to complete a matrix, but to entirely generate it.

CNN-SC is also pertinent to Neural Networks, which constitute nonlinear functions enabling the prediction of covariate responses in a flexible manner (Hastie, James, Witten, Tibshirani, & Taylor, 2013). These networks have found application in diverse domains including image classification and generation, forecasting, and medical research. Notably, they have been employed to generate contrast media states in Magnetic Resonance Imaging (MRI) scans of individuals afflicted with brain cancer, leveraging pre-contrast images (Bône & et al., 2022).

Section 1 of this paper introduces the topic, while Section 2 delves into the potential outcomes framework and addresses the problem of SUTVA violation. Section 3 establishes the connection between the potential outcome framework and kernels and convolutions. In Section 4, CNN-SC is explained in detail, covering its training procedure, architecture, and theoretical basis. Section 5 demonstrates the treatment effects that CNN-SC can estimate. Moving on to Section 6, the statistical inference framework is presented. Section 7 contains a Monte Carlo simulation comparing CNN-SC to DiD. Finally, a conclusion is provided in the last section.

# 2 Spatial problem setup

Consider the scenario where there are $i = 1, ..., N$ matrices observed over $t = 1, ..., T$ time periods, where $t_0$ represents the time when a spatial treatment occurs. Then, consider a dummy binary variable $D_i^{(j,k)}$ indicating if the matrix was directly affected by the treatment. Also, consider another dummy variable $D_{it}^{(j,k)} = 1_{(t>t_0)} \cdot D_i^{(j,k)}$ indicating if the matrix is affected by the treatment and is in the post-treatment period. This divides the matrices $N$ into a control group $N_0$ of unaffected matrices and -for simplicity- one (1) treated matrix $N_1$ affected by a spatial intervention. For simplification, suppose that there are only pre-treatment $t = 0$ and post-treatment period $t = 1$. Let $Y_{(i,t)}$ be a matrix of size $J \times K$ indexed by $i \in N$ on time $t \in T$ for a variable of interest. Also, lowercase $y_{(i,t)}^{(j,k)}$ represents the outcome of the raster's pixel in the $j$-th row and $k$-th column of the matrix $Y_{(i,t)}$ with $j, k \in (1, \dots, J) \times (1, \dots, K)$.

Therefore, the observed pixels' outcomes for the matrix $i$ at time $t$ can be represented as follows:

$$Y_{(i,t)} = \begin{pmatrix} y_{(i,t)}^{(1,1)} & y_{(i,t)}^{(1,2)} & \cdots & y_{(i,t)}^{(1,K)} \\ y_{(i,t)}^{(2,1)} & y_{(i,t)}^{(2,2)} & \cdots & y_{(i,t)}^{(2,K)} \\ \vdots & \vdots & \ddots & \vdots \\ y_{(i,t)}^{(J,1)} & y_{(i,t)}^{(J,2)} & \cdots & y_{(i,t)}^{(J,K)} \end{pmatrix}$$

On the other hand, each individual pixel in a matrix has its own treatment assignment. Denote by lowercase $d_{it}^{(j,k)} = 1_{(t>t_0)} \cdot d_i^{(j,k)}$ the dummy variable containing information of the direct treatment assignment of an individual pixel in the $j$-th row and $k$-th column of the $i$-th matrix at time $t$. This can be represented as the following matrix:[1]

$$D_{(i,t)} = \begin{pmatrix} d_{(i,t)}^{(1,1)} & d_{(i,t)}^{(1,2)} & \cdots & d_{(i,t)}^{(1,K)} \\ d_{(i,t)}^{(2,1)} & d_{(i,t)}^{(2,2)} & \cdots & d_{(i,t)}^{(2,K)} \\ \vdots & \vdots & \ddots & \vdots \\ d_{(i,t)}^{(J,1)} & d_{(i,t)}^{(J,2)} & \cdots & d_{(i,t)}^{(J,K)} \end{pmatrix}$$

An important clarification is that the pixel treatment status is independent of matrix treatment status only if the matrix is treated. That is, for the untreated matrices, the only possibility for every pixel treatment status is being untreated $d_{it}^{(j,k)} = 0; \forall t \in T, j \in J, k \in K$. However, for the treated matrices, there are two possibilities for the pixels' treatment status: $d_{it}^{(j,k)} = 0$ or $d_{it}^{(j,k)} = 1$, depending on the direct exposure to the spatial treatment.

## 2.1 Potential Outcomes Framework

In the present context, SUTVA implies that a pixel in a matrix should not have any effect on another pixel's outcome (Imbens & Rubin, 2015). However, in spatial settings, satisfying this assumption can be exceedingly challenging due to spatial autocorrelation and the presence of spillovers. Consequently, SUTVA violation undermines the Neuman-Rubin potential outcomes framework for this setting. To address this issue,

[1] It is important to note that the outcomes raster and the assignment matrix can be presented as a 3-dimensional array. For instance, the $y$ and $x$ axes represent the outcome rasters and the $z$ axis represents the treatment assignment of each matrix pixel. However, it is easier to explain as two 2-dimensional matrices.

this paper adopts the potential outcomes framework with interference proposed by Butts (2023) to better formalize the causal relations.

In this framework, the potential outcome for a matrix's pixel depends on the individual pixel's treatment status ($d_{it}^{(j,k)}$) and on a function $h_z^{(j,k)}$ that maps the overall treatment assignment matrix ($\vec{D}_i$) to a sparse matrix. Equation 1 presents the potential outcome for a given matrix's pixel.

$$y_{(i,t)}^{(j,k)}(d_{it}^{(j,k)}, h_z^{(j,k)}(\vec{D}_i)) \tag{1}$$

$h_z^{(j,k)}$ is referred to as the *exposure mapping function* as it maps the exposure of a pixel to the overall treatment status of the matrix. It is important to provide notation for a special case of exposure. For instance, if a unit is not exposed to any treated pixels, then it is denoted by $h_z^{(j,k)}(\vec{D}_i) = \vec{0}$. Here, $\vec{0}$ refers to a case where the pixel is not exposed to a treated pixel, and therefore not exposed to treatment. It is crucial to clarify that this does not imply that there is no interference; it only indicates that the interference does not contemplate treatment effect. Additionally, $h_z^{(j,k)}$ has a hyperparameter $z$ representing the degree to which a pixel is exposed to others within a matrix.

## 2.2 The Exposure Mapping Function

Characterizing the interference is important to explain how CNN-SC methodology addresses and tackles this issue. In this section, I provide a set of assumptions and explicitly define the *exposure mapping function*.

**Assumption 1:** *Stability of the pixels' exposure*

I assume that $h_z^{(j,k)}$ is stable for every time $t \in T$. This implies that the exposure for a given pixel to its neighbors does not vary across time. This assumption aids in identification as it allows us to characterize the exposure function using both pre-treatment and post-treatment data.

**Assumption 2:** *The interference follows a Chevyshev distance propagation*

Traditionally, the spatial autocorrelation literature assumes that the closer a unit is to others, the more affected it is by them. I flexibly follow this idea by considering that a pixel's potential outcomes depend on its neighbors up to some degree $z$, following a Chebyshev distance. However, I do not assume a distance-related

importance of its neighbors. As the next section will state, CNN-SC will flexibly estimate this relationship. As previously stated, $h_z^{(j,k)}(\vec{D}_i)$ transforms an assignment-to-treatment matrix to a sparse matrix considering only the $z$-th grade neighbors. Intuitively, for the first degree ($z = 1$), this metric will consider as neighbors the pixels directly connected to the analyzed one. Equation 2 formalizes this idea.

$$h_z^{(j,k)}(\vec{D}_i) = \begin{cases} d_{it}^{(j',k')} \text{ if } \{(j',k') \in \omega : \max\{|j-j'|, |k-k'|\} \leq z\} \\ 0 \quad \text{otherwise} \end{cases} \tag{2}$$

**Assumption 3:** *The interference is correctly identified within the matrix.*

An important assumption requires that the extent of a pixel's interference could be identified within the matrix. Formally:

$$\forall (j,k) : \exists z \in \mathbb{R}^+ \text{ with } z \leq J \text{ and } z \leq K$$

This implies that there exists a $z$-th degree neighborhood within the boundaries of the matrix. In practice, the matrix dimensions $J$ and $K$ could be reshaped to guarantee that the assumption holds. Additionally, this assumption may vary across the specific case of study. For example, when analyzing a raster with a grid size of 1000x1000 km, it could be argued that the $z$-th degree should be minor. However, with a grid size of 1x1 cm, then the $z$-th degree might be very large.

**Assumption 4:** *Spillovers are local*

Let $c([j,k],[j',k'])$ be the distance (Euclidean, Manhattan, Chebyshev, etc.) between a pixel $([j,k])$ in a matrix and one treated pixel $([j',k'])$. Then, this assumption states that pixels are no longer exposed to spillovers after some maximum distance to the treated pixels $\bar{c}$. This allows the existence of a control group that could provide a valid counterfactual for the absence of the spatial intervention or spillovers.

$$\min_{(j,k) \in i:D_i=1} \quad c([j,k],[j',k']) > \bar{c} \implies h_z^{(j,k)}(\vec{D}_i) = \vec{0}.$$

This assumption is purely theoretical, yet it is plausible to hold. For example, consider evaluating the effect of a policy on City A located in South America. Then, using City B in Europe as a control group ensures that Assumption 4 holds.

# 3 Convolution filters and Interference

A convolution filter relies on an operation called convolution, which corresponds to repeatedly multiplying matrix elements by another matrix element (kernel) and then adding the results into a scalar value (Hastie et al., 2013). In this section, I will argue that by using a convolution operation with a kernel, it is possible to estimate the interference as a weighted linear combination of the neighbors' outcomes. Additionally, I will discuss the trade-off between interpretability and prediction that arises from the use of kernels.

## 3.1 Convolution and Potential Outcomes Framework

To clarify the convolution operation, consider the following matrix (I) that simulates a raster with dimensions $6 \times 6$. Some outcomes are replaced with 0's to ease visualization. This does not alter the results. Additionally, a kernel ($K$) with a size of $3 \times 3$ is employed. Inside the matrix, the values are represented by a weight $w$ and the superindex represents the position of the weights in the matrix:

$$I_{6\times6} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & y^{2,2}_{(i,t)} & y^{2,3}_{(i,t)} & y^{2,4}_{(i,t)} & 0 & 0 \\ 0 & y^{3,2}_{(i,t)} & y^{3,3}_{(i,t)} & y^{3,4}_{(i,t)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad K_{3\times3} = \begin{pmatrix} w^{11} & w^{12} & w^{13} \\ w^{21} & w^{22} & w^{23} \\ w^{31} & w^{32} & w^{33} \end{pmatrix}$$

A convolution operation, denoted as $(I * K)$, represents the cross-product sum of two matrices. Here, $I(j,k)$ describes the value of the $j$-th row and $k$-th column of the raster, while $K()$ represents the position on the kernel:

$$(I * K)(j,k) = \sum_{j=0}^{3} \sum_{k=0}^{3} I(j,k) \cdot K(3-j, 3-k)$$

Expanding upon this operation, for a given position $(j,k)$, the convolution performs

a weighted linear combination:

$$(I * K)(j,k) = \sum_{j=0}^{3} \sum_{k=0}^{3} y_{(i,t)}^{j,k} \cdot w^{3-j,3-k} \tag{3}$$

In this context, convolution filters exhibit equivalence in the potential outcomes framework outlined in Section 2.1. Specifically, Equation 3 describes the outcome of a given pixel as a function of its own outcome ($y_{(i,t)}^{j,k}$) and the combination of the outcomes of the $z$-th degree neighbors ($h_z^{(j,k)}(\vec{D}_i)$), as detailed in Equation 1.

**Assumption 5:** *A unit's outcome interference can be expressed as the convex hull of its z-th degree neighbors' outcomes.*

Consider a raster image of size $J \times K$ and a kernel of size $M \times N$. Assume that the interference for a given pixel $(j,k)$ in the raster image is expressed as a linear weighted combination of the neighboring outcomes. For interpretability, I will enforce the condition that the weights must sum to one:

$$y_{(i,t)}^{(j,k)}(d_{it}^{(j,k)}, h_z^{(j,k)}(\vec{D}_i)) = \sum_{j=0}^{M} \sum_{k=0}^{N} y_{(i,t)}^{j,k} \cdot w^{M-j,N-k} \quad \text{s.t.} \quad \sum_{j=0}^{M} \sum_{k=0}^{N} w^{M-j,N-k} = 1 \tag{4}$$

After connecting the convolution operation with the potential outcomes framework, it is crucial to derive a data-driven method that allows for flexible and appropriate estimation of the kernel weights. In Section 4, I will present CNN-SC as a methodology to flexibly estimate the optimal weights that best fit the interference pattern.

## 3.2 Kernels: prediction vs interpretability

Section 3.1 connected the kernels and the convolution operation to the ability of aproaching to the interference of a pixel outcomes. However, it is important to discuss what is exactly identifyng one kernel when convolved. As expressed on equation 4 by finding the optimal weights $W*$ of the kernel $K$, I am finding the mean relation that a given pixel has with its neighbors. That is, it is implicitly characterizing the exposure mapping function $h_z^{(j,k)}(\vec{D}_i)$. However, by using only one kernel I am assuming that this function is common not only trough time, but also trough matrixes and pixels.

This assumption is binding as it is not real that the exposure function has to

be the same for different cells. For example, consider a raster with outcomes related to features such as trees, forests, rivers, among others. It is intuitive to think that the interference, the magnitudes and the related pixels for this features are specific and variant.

In this sense, I will sacrifice interpretability for a more accurrate prediction by using multiple kernels that follow the same idea exposed on section 3.1. The benefit of doing so is that the algorithm will have the possibility to flexible estimate a combination of optimal weights that may capture non-linearities and difference among features that will help to create a better counterfactual. At a glance, this allows to flexibly estimate $h_z^{(j,k)}(\vec{D}_i)$ without assuming that this function is common to all the matrixes and pixels.

# 4 Methodological proposal (CNN-SC)

## 4.1 Convolutional neural networks

CNNs are a subdivision of neural networks that efficiently process grid-like topology data. In general, multiple filters are convolved with the input to extract a representation of features in the topology. The optimal filters are learned during the training process, allowing for a data-driven approximation of the meaningful patterns of the image (Goodfellow, Bengio, & Courville, 2016). In this case, the interference of each pixel's potential outcomes. The proposed method consists of constructing a synthetic counterfactual untreated matrix $\hat{Y}_{it}(0)$ for the treated matrix. This joint estimation will create each counterfactual pixel for the matrix, providing an estimation for $\hat{Y}_{(i,t)}^{(j,k)}(d_{it}^{(j,k)}, h_z^{(j,k)}(\vec{D}_i))$.

For doing so, in the training stage, I will estimate using CNN a general function that maps for the untreated matrixes the pre-treatment raster to its post-treatment raster.

$$Y_{i,t+1} = \hat{f}(Y_{i,t} \mid D_{i,t} = 0) \tag{5}$$

By doing this, the generalized function $\hat{f}()$ learns the evolution of the confounders over entities. It also learns a representation of how spatially correlated pixels are affected among them, internalizing the way potential outcomes correlate. Then, the learned representation is applied to the treated unit $\hat{Y}_{it}(0) = \hat{f}(Y_{i,t} \mid D_{i,t} = 1)$, so the matrix prediction is obtained $\hat{Y}_{it}(0)$, as well as the individual pixel's prediction on absense of treatment $\hat{Y}_{(i,t)}^{(j,k)}(0, \vec{0})$.

## 4.2 Network Architecture and weights selection

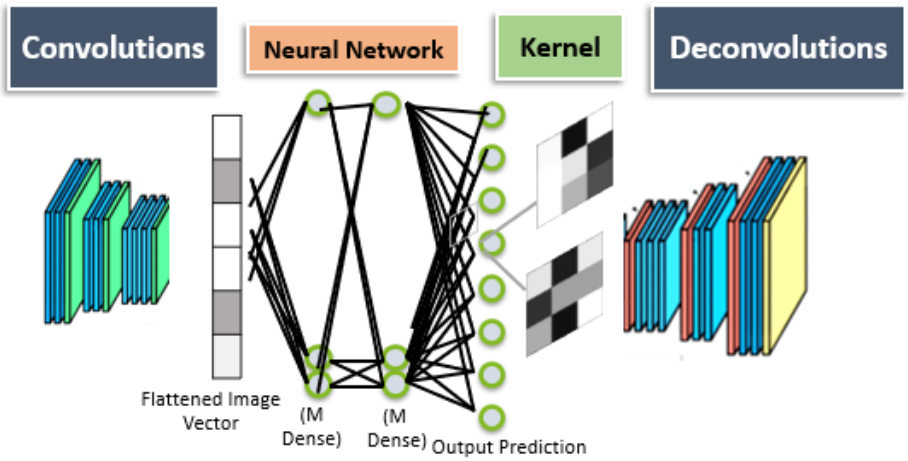
As expressed in Section 3.2, to improve prediction accuracy, CNN-SC will adopt a Convolutional Autoencoder architecture commonly used in image generation literature (Goodfellow et al., 2016). In the forward pass stage, a distribution of $N_0$ matrices of sizes $(J, K)$ is transformed through multiple convolutions of kernels of size $n \times m$, each denoted by $K_{n\times m}$, and a bias matrix or constant, denoted by $B_{n\times m}$, is added until a lower-rank latent representation of the matrix is achieved. This process, termed the decoder in Deep Learning literature, encodes the raster's information into a latent representation that can be easily processed by a neural network.

Subsequently, this representation is fed into a neural network that estimates the new matrix as follows, where $*$ represents the convolution operation:

$$Y_{i,t} = B_i + \sum_{m=1}^{n} Y_{i,t-1} * K_{i,m}$$

Afterward, I propose the use of deconvolutions to retrieve and decode the latent representation of the rasters, allowing the generation of a raster matrix of the original size. Figure 1 provides an example of the proposed network architecture.

Figure 2: Example of a convolutional encoder-decoder architecture for raster prediction




*Source*: Architecture based on Badrinarayanan et al. (2016)

After reconstructing a predicted image, an error measurement can be computed. Literature uses Mean Squared Error as a loss function for image prediction. This can be interpreted as the sum of the subtraction of the predicted and observed

matrix $E = \left(\sum_{i=1}^{N_1}(\hat{Y}_{i,t} - Y_{i,t})\right)^2$.

Backward propagation computes the directional derivatives of $E$ with respect to the kernel and biases following the equations $\frac{\partial E}{\partial K_{i,j}} = Y_{i,t-1} \cdot \frac{\partial E}{\partial Y_{i,t}}$ and $\frac{\partial E}{\partial B_i} = \frac{\partial E}{\partial Y_{i,t}}$. For the next propagation (epoch), the weights $K_{i,j}$ and $B_i$ are updated by moving towards the directional derivative, throughout a process called gradient descent with a defined learning rate. Finally, the learned function $\hat{f}()$ is then applied to the $N_1$ treated matrix.

# 5 Treatment Effects

CNN-SC generates a counterfactual matrix for a treated matrix. In this sense, Equation 6 presents the observed outcomes of the treated variable. On the other hand, Equation 7 presents the predicted counterfactual for the treated unit. As shown, CNN-SC estimates the outcome in the absence of treatment and exposure to treated units values for each pixel, denoted as $\hat{y}^{(j,k)}_{(i,t)}(0, \vec{0})$. Equation 6 also provides information regarding the treatment status of each individual pixel ($d^{(j,k)}_{it} = 1$ or $d^{(j,k)}_{it} = 0$).

$$Y_{(i,t)}|(D_{it} = 1) = \begin{pmatrix} y^{(1,1)}_{(i,t)}(d^{(1,1)}_{it}, h^{(1,1)}_z(\vec{D}_i)) & \cdots & y^{(1,3)}_{(i,t)}(d^{(1,3)}_{it}, h^{(1,3)}_z(\vec{D}_i)) \\ \vdots & \ddots & \vdots \\ y^{(j,1)}_{(i,t)}(d^{(j,1)}_{it}, h^{(j,1)}_z(\vec{D}_i)) & \cdots & y^{(j,k)}_{(i,t)}(d^{(j,k)}_{it}, h^{(j,k)}_z(\vec{D}_i)) \end{pmatrix} \tag{6}$$

$$\hat{Y}_{(i,t)}|(D_{it} = 1) = \begin{pmatrix} \hat{y}^{(1,1)}_{(i,t)}(0, \vec{0}) & \cdots & \hat{y}^{(1,3)}_{(i,t)}(0, \vec{0}) \\ \vdots & \ddots & \vdots \\ \hat{y}^{(j,1)}_{(i,t)}(0, \vec{0}) & \cdots & \hat{y}^{(j,k)}_{(i,t)}(0, \vec{0}) \end{pmatrix} \tag{7}$$

By subtracting both matrices $(Y_{(i,t)}|(D_{it} = 1) - \hat{Y}_{(i,t)}|(D_{it} = 1))$, it is possible to estimate the effect of the spatial intervention. This subtraction is performed for each entry of the matrix, such that for a pixel at the $j$-th and $k$-th position, the estimated treatment effect is $y^{(j,k)}_{(i,t)}(d^{(j,k)}_{it}, h^{(j,k)}_z(\vec{D}_i)) - \hat{y}^{(j,k)}_{(i,t)}(0, \vec{0})$. The treatment effect can be summarized in another $J \times K$ matrix, as presented in Equation 8.

$$\hat{\tau}_{(i,t)}|(D_{it} = 1) = \begin{pmatrix} \hat{\tau}^{(1,1)}_{(i,t)} & \cdots & \hat{\tau}^{(1,3)}_{(i,t)} \\ \vdots & \ddots & \vdots \\ \hat{\tau}^{(j,1)}_{(i,t)} & \cdots & \hat{\tau}^{(j,k)}_{(i,t)} \end{pmatrix} \tag{8}$$

## 5.1 Treatment Effects over the Pixels

It is important to note that the treatment effect over a matrix estimates different effects depending on the treatment assignment of a pixel ($d_{it}^{(j,k)}$)[2]. CNN-SC can estimate two different effects: i) The total effect on a pixel, and ii) the spillover effect on the untreated pixels.

**The Total Effect over the Treated Pixels:** This effect is obtained when turning on treatment for a pixel and moving from zero exposure to treatment to an exposure to the treated units given by the exposure mapping function: $h_z^{(j,1)}(\vec{D}_i)$. Equation 10 presents the estimation of this effect.

$$\hat{\tau}_{\text{total},(i,t)}^{(j,k)} = y_{(i,t)}^{(j,k)}(1, h_z^{(j,k)}(\vec{D}_i)) - \hat{y}_{(i,t)}^{(j,k)}(0, \vec{0}) \tag{9}$$

By adding and subtracting $\hat{y}_{(i,t)}^{(j,k)}(1, \vec{0})$, it is possible to identify that the total effect is an additive combination of the direct effect of the treatment and the spillovers on the treated[3].

$$\hat{\tau}_{\text{total},(i,t)}^{(j,k)} = \hat{\tau}_{\text{direct},(i,t)}^{(j,k)} + \hat{\tau}_{\text{spill},(i,t)}^{(j,k)}(1) \tag{10}$$

This effect allows us to understand the direct consequences of the spatial intervention on the analyzed outcomes, providing an estimate of interest for public policy.

**Spillovers on the Untreated Pixels:** On the contrary, consider an untreated pixel in the treated matrix. As described, in the presence of spillovers, the literature created geometrical ways of exploring and estimating the spillover effects, their magnitude, and their propagation. A key contribution of CNN-SC is the possibility to flexibly estimate and identify the spillover propagation and magnitudes. This is done by not assuming any geometrical propagation, nor assuming that closer units will be affected differentially. In this sense, the method provides a data-driven approximation for the identification of spillovers.

Essentially, for the untreated pixels in a treated matrix, the method estimates the spillover effect ($\hat{\tau}_{\text{spill}}(0)$), where 0 represents the spillover on the untreated.

[2]This depends entirely on the data and the specific application, as in a matrix every pixel could be directly treated.

[3]In the spatial intervention literature, the discussion of spillovers on the treated accepts that there is an intensity or differential on the outcomes if units are exposed to other treated units. It can be seen as the hidden variations of treatment explained by Imbens and Rubins, 2015

As shown in Equation 11:

$$\hat{\tau}^{(j,k)}_{\text{spill},(i,t)}(0) = y^{(j,k)}_{(i,t)}(0, h^{(j,k)}_{z}(\vec{D}_i)) - \hat{y}^{(j,k)}_{(i,t)}(0, \vec{0}) \tag{11}$$

Intuitively, the equation subtracts the observed outcome - presumably contaminated by the treatment - from the predicted counterfactual in the absence of any treated pixels in the matrix. This difference is argued to be the spillover effects.

**Estimating Noise:** Finally, due to the fact that spillover and contaminated pixels are not known a priori to the investigator, it is possible that when using CNN-SC, we estimate a noise effect. Consider the effect described in Equation 12.

$$\hat{\tau}^{(j,k)}_{\text{noise},(i,t)}(0) = y^{(j,k)}_{(i,t)}(0, \vec{0}) - \hat{y}^{(j,k)}_{(i,t)}(0, \vec{0}) \tag{12}$$

This equation states that it is possible that for an untreated pixel, there is no spillover; yet, due to the function's irreducible error, it is possible to estimate a treatment effect for a pixel. I will further prove that even though there might be an estimated noise effect for an individual pixel, on average, this error converges to 0. For the moment, I will assume that.

**Assumption 6:** *Average Noise Estimates Converge to 0*

$$E[\hat{\tau}^{(j,k)}_{\text{noise},(i,t)}(0)] \implies 0$$

## 5.2 Generalizing the Treatment Effects

Section 5.1 presented the individual effects over the pixels that CNN-SC is able to identify. However, there are some aggregated effects that are of interest to investigators. By aggregating, it is possible to identify the following effects: the Average Treatment effect on the Treated (ATT), the Average Spillover Effect (ASE), and the Average Grid Effect (AGE).

**The Average Treatment Effect on the Treated (ATT):** This effect is computed after averaging through all the total effects ($\hat{\tau}^{(j,k)}_{\text{total},(i,t)}$) on the treated pixels present in the matrix. This effect is very informative, as it stands for the average

total effect that the spatial treatment had on the interest outcome.

$$ATT_t = \sum_{j=0}^{J} \sum_{k=0}^{K} \hat{\tau}^{(j,k)}_{total,(i,t)} \cdot d^{(j,k)}_{it} \tag{13}$$

**The Average Spillover Effect (ASE):** This effect is computed after averaging through all the estimated effects for untreated pixels. This effect is of major importance. For example, consider a crime-related intervention such as assigning police officers to a location. Arguably, if the raster outcome is a crime-related variable such as homicides, the total effect on the treated pixels might indicate a reduction of crime. But, it could be possible that when accounting for the spillover effects, the movement of the crime from that location increased the overall crime. Thus, this effect is vital to understanding and evaluating the overall impact of a spatial intervention, especially when comparing the ASE to the ATT.

$$ASE_t = \sum_{j=0}^{J} \sum_{k=0}^{K} \hat{\tau}^{(j,k)}_{total,(i,t)} \cdot (1 - d^{(j,k)}_{it}) \tag{14}$$

It is important to identify that Equation 15 can be expressed as:

$$ASE_t = E[\hat{\tau}^{(j,k)}_{noise,(i,t)}(0)] + E[\hat{\tau}^{(j,k)}_{spill,(i,t)}(0)]$$

By Assumption 6: $ASE_t = E[\hat{\tau}^{(j,k)}_{spill,(i,t)}(0)]$. This is a very strong result, as it states that investigators do not have to know a priori the distribution of the contaminated pixels to properly identify the spillover effects. In other proposals in the literature, there are a priori assumptions of the propagation of the spillovers (i.e., rings and distances).

**The Average Grid Effect (AGE):** The Average Grid Effect is informative of the overall spatial treatment effect. It is easy to show that it is a weighted combination of the ATT and ASE.

$$AGE_t = \sum_{j=0}^{J} \sum_{k=0}^{K} \hat{\tau}^{(j,k)}_{total,(i,t)} \tag{15}$$

This effect is of particular importance for doing the statistical inference, as it will be described in section 6.

### 5.3 Internal vs External Validity of the Methodology

CNN-SC grants internal validity of the experiment; however, due to the fact that the overall treatment assignment is specific to the spatial intervention, external validity is not guaranteed. To better explain this, the method is estimating for a given pixel $y^{(j,k)}_{(i,t)}(d^{(j,k)}_{it}, h^{(j,k)}_{z}(\vec{D}_i)) - \hat{y}^{(j,k)}_{(i,t)}(0, \vec{0})$. But, for slightly different observed treatment assignments ($\vec{D}_i$), the treatment effect does not provide information on the change of the overall treatment status of the matrix. In this sense, the external validity of the method is only plausible when comparing it to spatial interventions with a similar treatment setup structure.

Mathematically, this can be perceived by the fact that the total effect is a combination of the direct effect and the spillovers on the treated pixels (see Equation 10). In this sense, to provide external validity it is needed that $\hat{\tau}^{(j,k)}_{spill,(i,t)}(1) = 0$. In other words, for a treated pixel, there are no variations in the treatment intensity changing by the exposure to a different amount of treated units (Imbens & Rubin, 2015).

## 6 Statistical inference

Providing valid Statistical Inference for this methodology is crucial to understand the causal relations between the spatial intervention and the outcomes. I will use a classical placebo inference test common in the literature of Synthetic Controls and Randomization Inference (Heß, 2017; Abadie & Gardeazabal, 2003).

### 6.1 Test setup

In the synthetic control literature, a common way of addressing placebo tests is to estimate a synthetic control for every untreated unit and then analyzing how atypical the treated unit's estimated effect is when compared to the placebo ones. However, CNN-SC is a very computationally intensive method. Therefore, I will keep the original idea but do it in a computationally efficient way.

Randomly separate the $N_0$ ($D_i = 0$) matrices into three groups: A training sample (TS), a Validation sample (VS), and a Placebo sample (PS). Use both TS and VS to train the algorithm as presented in Equation 5.[4] Afterwards, apply the learned function $\hat{f}()$ to both the PS and the $N_1$ ($D_i = 1$) matrix. Then, estimate the

[4]Specially, to tune the hyperparameters such as the weights of the CNN

$AGE_i$ for each individual matrix as explained in Equation 15. Also, define $AGE_i^{N_1}$ as the Average Grid Effect of the original treated matrix.

### 6.2 Statistical Test

I will test the null hypothesis that there is no treatment effect for any grid. On the contrary, the alternative hypothesis states that there exists a matrix for which there is a treatment effect (i.e., the $N_1$ matrix).

$$Hypothesis = \begin{cases} H_0 : AGE_{(t,i)} = 0 & \forall i \in (PD \bigcup N_1) \\ H_a : AGE_{(t,i)} \neq 0 & \exists i \in (PD \bigcup N_1) \end{cases}$$

The associated p-value is then computed as the proportion of estimated average grid effects that are greater than the originally treated matrix. Intuitively, this is a Granger causation test. If for an untreated and unrelated unit the method estimates a greater effect than for the actual treated unit, then it seems that the results were obtained by "chance".

$$p - value = \frac{\sum_{i=1}^{\|PD\|} 1\left(\left|AGE_{(t,i)}\right| > \left|AGE_t^{(N_1)}\right|\right)}{\|PD\|} \tag{16}$$

Equation 16 is counting the number of times that the absolute placebo AGEs are greater than the absolute AGE for the treated matrix, using an indicator function. Then, it divides them by the number of placebo matrices considered. Finally, the investigator can reject the null hypothesis based on a level of significance if $\alpha > p - value$.

## 7 Monte Carlo Simulations

Due to the fundamental problem of causal inference, we are not able to observe all the potential outcomes for a pixel. This makes difficult to test the godness of a methodology. As a solution, it is common in the literature to simulate a data distribution and test various methodologies within the controlled framework.

### 7.1 Simulating confounders and the spatial interference

In this section, I will simulate a spatial intervention. I consider 100 matrices, each one with a common size of 10x10 pixels. 99 matrices will be untreated ($D_i = 0$) and one matrix will be treated for a specific pixel ($D_i = 1$). For each matrix, there will

be two time periods $t \in \{0, 1\}$: pre-treatment and post-treatment. I will simulate a confounder evolution, the spatial interference, the treatment effect, and the spillovers. The simulation procedure is visually explained in Figure 3.

First, for the pre-treatment values ($t = 0$), I will start by simulating an i.i.d outcome following a uniform distribution, however, varying its parameters across the treated and untreated matrices. This will create an average difference in the outcomes across subgroups and will test the methodology's ability to learn the evolution of outcomes.[5]

$$y_{(i,0)}^{(j,k)} = \begin{cases} y_{(i,0)}^{(j,k)} \sim \mathcal{U}(0, 235) \text{ if } D_i = 0 \\ y_{(i,0)}^{(j,k)} \sim \mathcal{U}(26, 255) \text{ if } D_i = 1 \end{cases}$$

After creating this i.i.d sample, I simulate the interference of a $z$ degree neighborhood for every pixel in every matrix. This is done by replacing each $y_{(i,0)}^{(j,k)}$ pixel by the mean value of the 1st degree neighbors' outcomes. Formally, replacing the values by the result of this operation:

$$y_{(i,0)}^{(j,k)} = \sum_{m=0}^{3} \sum_{n=0}^{3} \frac{y_{(i,0)}^{j-m,k-n}}{3 * 3} \tag{17}$$

The *Spatial Interference algorithm* is as follows. It starts with the position (1,1) of the matrix and applies Equation 17 to the value of the 1-st row, 1-st column pixel. Then it iteratively replaces the outcomes for the cells first by moving through the columns and keeping the 1-st row fixed. After that, it goes to the second column and moves again through all the columns. This iteratively repeats until it replaces the $j$-th and $k$-th pixel of the matrix. The procedure is common to all matrices, and the propagation can be visually perceived in Figure 3.

Second, the post-treatment values of each matrix are simulated by adding confoundedness using an unobserved variable that comes from a normal distribution with mean 40 and variance 2. Also, a treatment effect with a value of 10 is added to specifically four pixels in the treated matrix: the ones present on the 5th and 6th row, and 5th and 6th column.

$$y_{(i,1)}^{(j,k)} = \begin{cases} y_{(i,1)}^{(j,k)} + N(40, 2) + 10 & \text{if } j = [5, 6], k = [5, 6] \text{ and } D_i = 1 \\ y_{(i,1)}^{(j,k)} + N(40, 2) & \text{otherwise} \end{cases}$$

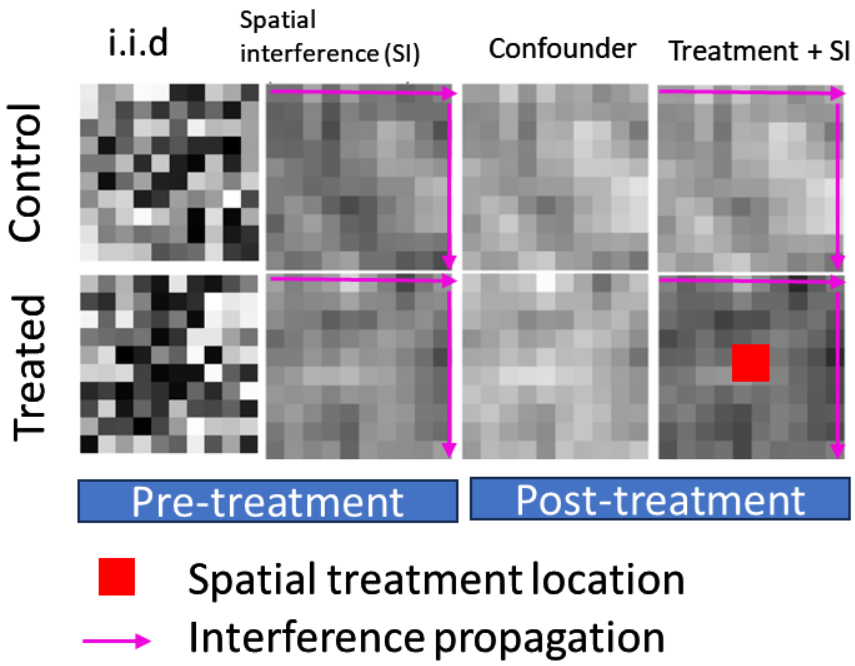
[5]The values ranging from 0 to 255 are selected due to their white, gray, and black representation.

Besides, the interference is created again by replacing each pixel's outcome $y_{(i,1)}^{(j,k)}$ by the mean value of the 1st degree neighbors' outcomes. Following the *spatial interference algorithm* previously explained, but, after the confounding and treatment effect on the outcome.

$$y_{(i,1)}^{(j,k)} = \sum_{m=0}^{3} \sum_{n=0}^{3} \frac{y_{(i,1)}^{j-m,k-n}}{3*3}$$

Note that for the treated matrix, this also generates a spillover effect, as the treatment affects some untreated pixels, specifically their first-degree neighbors. It is crucial to emphasize that due to the specific nature of the spatial interference algorithm, the spillovers propagate in a defined manner. Since the algorithm starts with the (1,1) pixel and only considers the first-degree neighbors, these spillovers will propagate exclusively to pixels with positions greater than $j = 4$ and $k = 4$. Pixels located before this threshold only experience interference from other units but are not affected by the treatment during the mean value normalization process. By simulating the interference in this manner, I aim to test and demonstrate the method's ability to accurately estimate not only the spillovers but also to effectively identify their propagation pattern.

Figure 3: visualization of the simulation exercise



## 7.2 Comparing methods

Given the panel data structure and the presence of interference and spillovers, it's crucial to employ methodologies that appropriately account for these factors when conducting causal inference. To ensure a fair comparison with CNN-SC, which explicitly addresses spillovers, I will utilize the extended Difference in Differences (DID) framework proposed by Butts (2023). Traditional DID and Synthetic Control methodologies do not inherently incorporate spillovers, making a comparison unfair.

Adapting Butts (2023) methodology to the raster matrices problem involves estimating the following equation:

$$y_{(i,1)}^{(j,k)} - y_{(i,0)}^{(j,k)} = \alpha + \tau d_{(i,1)}^{(j,k)} + \sum_{z} \alpha_z * ring_{(i,z)}^{(j,k)} + (\epsilon_{(i,1)}^{(j,k)} - \epsilon_{(i,0)}^{(j,k)}) \tag{18}$$

where $d_{(i,1)}^{(j,k)}$ is a dummy variable taking the value of 1 if the individual pixel was treated and 0 if else. Also, $ring_{(i,z)}^{(j,k)}$ is a set of dummy variables that takes the value of 1 if the j-th, k-th pixel is a neighbour in a degree z to a treated unit, 0 otherwise. It is important to state that the average values of the spillover estimates ($E(\alpha_z)$) has a correspondance with the ASE effect proposed on equation 15 for CNN-SC. Finally, $\epsilon_{(i,0)}^{(j,k)}$ indicates an idiosincratic error specific to each cell. In this case, note that this error includes the unobserved confounder simulated previously.

On the other hand, I estimate the pixel's effects using CNN-SC. As mentioned on section 4, I train a convolutional autoencoder using the untreated matrixes pre-treatment pixels to predict its post-treatment states $Y_{i,1} = \hat{f}(Y_{i,0} \mid D_{i,1} = 0)$. Then, this learned latent representation is applied to the treated matrix pre-treatment state to predict the counterfactual raster $\widehat{Y_{i,1}} = \hat{f}(Y_{i,0} \mid D_{i,1} = 1)$.

## 7.3 Preliminary Results

Table 1 presents the results of comparing the treatment effect estimation for DiD and the proposed method for 10 simulations of the previously proposed Monte Carlo simulation. Upon comparing the results to the simulated treatment value (10), it is evident that CNN-SC performed the best, with a deviation of 13% from the original value. This deviation is notably lower than that of the other methods. For instance, DiD, accounting for spillovers, exhibited a deviation of 28%.

Table 1: Estimated effects comparison

| | **DID** | **DID-Spillovers** | **CNN-SC** |
|---|---|---|---|
| ATT ($\tau$) | 5.4 | 7.2 | 11.3 |
| ASE | - | - | 5.2 |
| $E(\alpha_z)$ | - | 3.8 | - |
| Observations | 10000 | 10000 | 100 |

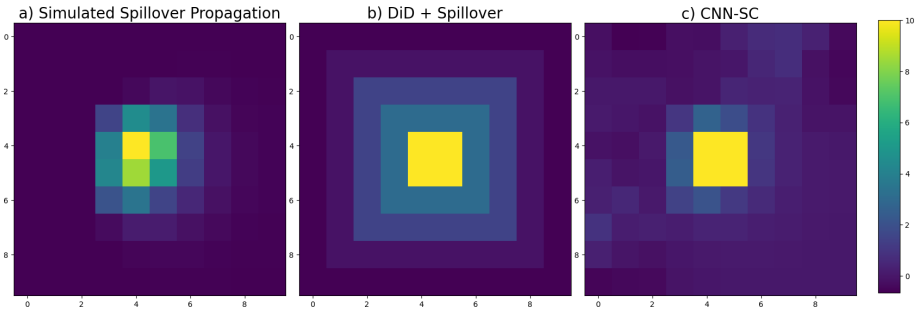
*Notes*: Estimated effects are the mean computed effects for 10 repetitions of the monte carlo simulation. For a grid of size 10x10, the maximum neighborhood degree z for the matrix is 10, where the $\alpha_{10}$ coefficient is an estimate of the spillover effects for the matrix's corner.

Figure 4 presents the spillover propagation and identification by comparing two methodologies. Panel (a) describes the simulated propagation, showcasing two key elements. Firstly, the observed propagation occurs diagonally as previously described. Secondly, it illustrates the concept of spillovers over the treated units. Notably, spillovers affecting treated units vary across pixels, particularly due to neighboring treatment assignments.

Panel (b) displays the estimation of the Difference-in-Differences (DiD) method with spillovers, employing the rings estimation for spillover identification proposed by Butts (2023). The results reveal constant values for pixels within a radius distance of the treated pixels. However, a notable concern is that it estimates mean values of spillover effects for pixels not exposed or contaminated by the treatment.

Panel (c) presents the estimation for CNN-SC. It depicts noise treatment effects for untreated and uncontaminated pixels, notably observed in the first rows and columns. However, CNN-SC accurately identifies the propagation of spillovers, evidenced by varying main effects primarily concentrated in the bottom right pixels in a diagonal propagation. As simulated. This also shows the preliminary ability of CNN-SC to undercover spillovers whithout assuming a propagation distribution.

Figure 4: Spillover propagation preliminary results




*Notes*: Author's calculation using the simulation procedure described on section 7. The figure presents the simulated spillover propagation, the constant estimates for DiD with spillovers methodology proposed by Butts (2023), and CNN-SC. The direct effect (10) was replaced for the treated squared region for both methodologies, this is done to focus the discussion on the spillovers over the untreated. The figure presents the visualization for 1 of the 10 repetitions of the simulation.

# 8 Concluding Remarks

I explain a methodology called CNN-SC that explores the application of convolutional neural networks for spatial causal inference, particularly in the context of estimating treatment effects and spillovers in raster-based data. I present the convolution operation and argue its ability to represent the exposure mapping function presented on the potential outcomes framework with spillovers. I also show the method's ability to estimate interference as a weighted linear combination of neighboring outcomes. I provide the networks architecture and examine how to apply it.

In the text, I also present its capacity to estimate treatment effects over pixels such as the total effect and the spillovers effect. Also, on how to aggregate the effects to identify the Average Treatment effect on the Treated (ATT), Average Spillover Effect (ASE), and Average Grid Effect (AGE). Besides, I discuss the placebo analysisis for statistical inference. As stated, the method has two advantages. First, it is capable of identify spillovers without explicitly defining the propagation of the spillovers. Second, it allows to estimate causal effects under the presence of interference, wich usually violates SUTVA and denies valid inference.

Furthermore, I present a simulation framework to test the method's performance in scenarios with spatial interference and spillovers, comparing it to established methodologies like Differences in Differences. Ultimately, this paper underscores

the potential of convolutional neural networks in spatial causal inference, offering a promising avenue for further research on analyzing complex spatial interventions and their spillovers propagation.

# References


Abadie, A., & Gardeazabal, J. (2003). The economic costs of conflict: A case study of the basque country. *American Economic Review*, *93*(1), 113–132.

Athey, S., Bayati, M., Doudchenko, N., Imbens, G., & Khosravi, K. (2021). Matrix completion methods for causal panel data models. *Journal of the American Statistical Association*, *116*(536), 1716–1730. doi: 10.1080/01621459.2021.1891924

Athey, S., & Imbens, G. (2019). Machine learning methods that economists should know about. *Annual Review of Economics*, *11*, 685–725.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2016). *Segnet: A deep convolutional encoder-decoder architecture for image segmentation.*

Brodersen, K. H., Koehler, J., Remy, N., Scott, S. L., & Gallusser, F. (2015). Inferring causal impact using bayesian structural time-series models. *Annals of Applied Statistics*, *9*, 247–274.

Butts, K. (2023). *Difference-in-differences estimation with spatial spillovers.*

Bône, A., & et al. (2022, Aug). From dose reduction to contrast maximization: Can deep learning amplify the impact of contrast media on brain magnetic resonance image quality? a reader study. *Invest Radiol*, *57*(8), 527–535. doi: 10.1097/RLI.0000000000000867

Doudchenko, N., & Imbens, G. (2016). Balancing, regression, difference-in-differences and synthetic control methods: A synthesis.

Fick, S. E., Nauman, T. W., Brungard, C. C., & Duniway, M. C. (2021). Evaluating natural experiments in ecology: using synthetic controls in assessments of remotely sensed land treatments. *Ecological Applications*, *31*(3), 1–16. Retrieved from https://www.jstor.org/stable/27029215

Gonzalez, R. M. (2021). Cell phone access and election fraud: Evidence from a spatial regression discontinuity design in afghanistan. *American Economic Journal: Applied Economics*, *13*(2), 1–51. doi: 10.1257/app.20190443

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* MIT Press.

Hastie, T., James, G., Witten, D., Tibshirani, R., & Taylor, J. (2013). *An introduction to statistical learning* (1st ed.). Springer.

Heß, S. (2017). Randomization inference with stata: A guide and software. *The Stata Journal*, *17*(3), 630-651. doi: 10.1177/1536867X1701700306

Huber, M. (2021). *Causal analysis: impact evaluation and causal machine learning with applications in r.* (https://drive.switch.ch/index.php/s/tNhKQmkGB48bjfz)

Imbens, G. W., & Rubin, D. B. (2015). *Causal inference for statistics, social, and biomedical sciences: An introduction.* Cambridge University Press.

Keele, L. (2015). The statistics of causal inference: A view from political methodology. *Polit. Anal.*, *23*(3), 313–335.

Pollmann, M. (2022). Causal inference for spatial treatments. (Working Paper)

Ruining, J., Shuai, S., & Lili, Y. (2021). High-speed rail and co2 emissions in urban china: A spatial difference-in-differences approach. *Energy Economics*, *99*.

Sims, K., & Alix-Garcia, J. M. (2017). Parks versus pes: Evaluating direct and incentive-based land conservation in mexico. *Journal of Environmental Economics and Management*, *86*(C), 8-28. Retrieved from https://EconPapers.repec.org/RePEc:eee:jeeman:v:86:y:2017:i:c:p:8-28