Journal of Statistical Software

September 2025, Volume 114, Issue 7. doi: 10.18637/jss.v114.i07

TSCI: Two Stage Curvature Identification for Causal Inference with Invalid Instruments in R

David Carl
ETH Zürich

Corinne Emmenegger
ETH Zürich

Peter Bühlmann
ETH Zürich

Zijian Guo
Rutgers University

Abstract

TSCI implements treatment effect estimation from observational data under invalid instruments in the R statistical computing environment. Existing instrumental variable approaches rely on arguably strong and untestable identification assumptions, which limits their practical application. **TSCI** does not require the classical instrumental variable identification conditions and is effective even if all instruments are invalid. **TSCI** implements a two-stage algorithm. In the first stage, machine learning is used to cope with nonlinearities and interactions in the treatment model. In the second stage, a space to capture the instrument violations is selected in a data-adaptive way. These violations are then projected out to estimate the treatment effect.

Keywords: endogeneity, instrumental variables, nonparametric treatment model, treatment effect, R.

1. Introduction: Invalid instruments

Inferring causal treatment effects from observational studies may suffer from endogeneity due to unmeasured confounders. A common remedy is to use instrumental variables (IVs) to isolate the variation in the treatment that is uncorrelated with the unmeasured confounders. However, valid inference requires these IVs to satisfy stringent and untestable assumptions, i.e., assumptions A2–A3 below. Conditioning on the based covariates, the IVs

- Assumption A1: need to be associated strong enough with the treatment variable;
- Assumption A2: must not be associated with the unmeasured confounders;
- Assumption A3: must not directly affect the outcome variable.

In practice, assumptions A2 and A3 may be questionable or uncheckable, and empirical analyses often rely on external knowledge to verify them, which may be prone to errors. Therefore, it is crucial to develop methods that are agnostic to these assumptions because falsely relying on them may introduce substantial bias and invalidate inference. We consider IVs that violate assumptions A2 or A3 and call such IVs *invalid*. To cope with invalid IVs, the following approaches with software for the R environment for statistical computing (R Core Team 2025) exist.

- Assuming near-orthogonality of the effect of the IVs on the treatment and the direct effect of the IVs on the outcome, for which software is available in the supplementary material of Bowden, Davey Smith, and Burgess (2015).
- Assuming a treatment model with heteroscedastic errors, for which software is available on the GitHub repository of Tchetgen Tchetgen, Sun, and Walter (2021).
- Selecting valid IVs from a pool of potentially invalid ones, for which software is available in the R packages **RobustIV** (Koo, Wang, and Guo 2022b) implementing robust causal inference (Guo, Kang, Cai, and Small 2018; Guo 2023), **controlfunctionIV** (Koo, Li, and Guo 2022a) implementing a control function approach (Li and Guo 2020), and **CIIV** (Windmeijer and Liang 2021) implementing the confidence interval method (Windmeijer, Liang, Hartwig, and Bowden 2021). See Koo, Lee, Small, and Guo (2023) for the implementation details of **RobustIV** and **controlfunctionIV**.

An implementation of IV selection can also be found in the Stata (StataCorp 2025) module **SIVREG** (Farbmacher 2018). Please also see Guo and Bühlmann (2022) for more references to these approaches.

The two stage curvature identification (TSCI) approach proposed by Guo and Bühlmann (2022) makes none of the restrictions A1–A3, and all IVs may be invalid. The rather mild key assumption is that violations and the association between the IV and the treatment arise from different functional forms. That is, coincidental cases where they are all, for instance, linear, are excluded. Machine learning is used to learn the possibly nonlinear treatment model. Our R package, **TSCI**, provides software for this method and is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=TSCI`.

For valid IVs, R packages exist to estimate treatment effects from observational data using machine learning, **DoubleML** (Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins 2018; Bach, Kurz, Chernozhukov, Spindler, and Klaassen 2024), and machine learning with additional regularization, **dmlalg** (Emmenegger and Bühlmann 2021; Emmenegger 2022). However, they use a linear fitting between the IV and the treatment. To incorporate nonlinearities between the IVs and the treatments, Fan and Zhong (2018) proposed a nonparametric additive model for the treatment model, which is implemented in the R package **naivereg** (Fan, He, and Zhong 2020). In contrast, **TSCI** can cope with invalid instruments, and it uses machine learning to capture complex nonlinearities and interaction terms among the IVs and covariates in the treatment model.

In the overidentification regime with more instruments than treatments, the Sargan test is useful in testing instruments' validity. The Sargan test has been implemented in the standard IV software, such as the R package **ivmodel** (Kang, Jiang, Zhao, and Small 2021, 2023). However, **TSCI** provides validity for the just-identification regime by leveraging the nonlinearity of the treatment model.

This paper is organized as follows. Section 2 presents the statistical model and discusses key steps of the **TSCI** algorithm on a theory level. Section 3 explains how to use **TSCI** to estimate and make inference for the treatment effect from observational data under invalid IVs using empirical data. Section 4 summarizes **TSCI**'s main functionality, and Section 5 finally concludes.

2. Two stage curvature identification

Subsequently, we present the TSCI methodology of Guo and Bühlmann (2022). We have $i = 1, \ldots, n$ independent and identically distributed observations as follows. For IVs Z_i and exogenous baseline covariates X_i, we consider the treatment model

$$D_i = f(Z_i, X_i) + \delta_i, \quad \mathsf{E}[\delta_i \mid Z_i, X_i] = 0 \tag{1}$$

for some unknown function f, and we consider the outcome model

$$Y_i = \beta D_i + g(Z_i, X_i) + \epsilon_i, \quad \mathsf{E}[\epsilon_i \mid X_i, Z_i] = 0$$

for some unknown function g. The error terms δ_i and ϵ_i may be correlated, which introduces endogeneity. If the IVs Z_i were valid, the function g would not directly depend on Z_i. We aim to estimate and make inference for the effect β of the treatment on the outcome. The variables X_i and Z_i may be multi-dimensional. All other variables, and thus also β, are 1-dimensional. Introducing $\phi(X_i) = \mathsf{E}[g(Z_i, X_i) \mid X_i]$, we can rewrite the outcome model as

$$Y_i = \beta D_i + h(Z_i, X_i) + \phi(X_i) + \epsilon_i, \quad \mathsf{E}[\epsilon_i \mid X_i, Z_i] = 0 \tag{2}$$

with $h(Z_i, X_i) = g(Z_i, X_i) - \mathsf{E}[g(Z_i, X_i) \mid X_i]$. If the IVs Z_i were valid, we had $h \equiv 0$ because g would not directly depend on Z_i. Conversely, if h is non-zero, then assumptions A2 or A3 are violated. Consequently, h is a measure of IV violation, and we call it *violation function*.

The crucial idea of TSCI to estimate β is to make use of differences in the functional forms of the treatment model f and the violation function h. Assume that f can be approximated well by some subspace $\mathcal{S}$ of some Hilbert space S and that $g = h + \phi$ can be approximated well by some subspace $\mathrm{Span}(\mathcal{V}) \subseteq S$ spanned by some set of basis functions $\mathcal{V}$. To cope with the IV violation, we need to project out from the outcome model the part of g that is "non-orthogonal" to f. That is, it is not necessary to project out all of g, thus all of $\mathrm{Span}(\mathcal{V})$, but it suffices to project to a space where f and g become "orthogonal". That is, we choose some space $\mathcal{T}$ satisfying $\mathcal{S} \cap \mathrm{Span}(\mathcal{V}) \subseteq \mathcal{T} \subseteq S$, and we will project to the orthogonal complement of $\mathcal{T}$. We call $\mathcal{T}$ *violation space*, and an example of such a $\mathcal{T}$ is $\mathrm{Span}(\mathcal{V})$ itself. Because $\mathcal{T}$ is unknown in practice, **TSCI** selects one out of user-specified candidates in a data-adaptive way.

Provided enough variation in the approximation of f using $\mathcal{S}$ remains after projecting it onto the orthogonal complement of $\mathcal{T}$, we can use these projected values to overcome the IV violation. In particular, if h is linear in Z_i and f is quadratic in Z_i, we could use the nonlinear part of f in Z_i to estimate β. Note that in Guo and Bühlmann (2022), $\mathcal{T}$ is not explicitly defined but is chosen as $\mathrm{Span}(\mathcal{V})$.

To estimate the treatment effect β, we require approximating two objects: the function f and the violation space $\mathcal{T}$, which may depend on the method chosen to approximate f. The

function f is estimated using machine learning (or a basis expansion), $\mathcal{T}$ is selected by testing user-specified candidates for which the resulting (generalized) IV strength is large enough. In particular, the candidates differ in the basis used to approximate h but contain the same basis to approximate ϕ. The TSCI algorithm consists of the following two stages:

- Using machine learning to predict the treatment in the treatment model in Equation 1.
- Rescaling the outcome by a "hat matrix" that comes from the above prediction step, projecting the rescaled response and predicted treatment to the orthogonal complement of the violation space (where IV violation is no longer a problem), and performing ordinary least squares.

An additional data splitting strategy is required if machine learning is used to approximate f. In the classic two stage least squares (TSLS) algorithm (Theil 1953b,a), the responses need no rescaling in the second step. **TSCI** requires it because the "hat matrix", in contrast to TSLS, may not be orthogonal. The next sections detail sample splitting, the basis expansion of ϕ, and the estimation of Span($\mathcal{V}$), f, and finally β. Subsequently, we denote in bold row-wise concatenations of the observations; for instance, $\mathbf{h}$ represents the vector $(h(Z_1, X_1), \ldots, h(Z_n, X_n))^\top$.

2.1. Sample splitting and machine learning estimator of f

If a flexible machine learning algorithm such as random forest or boosting is used to fit the treatment model, sample splitting is essential to remove bias due to overfitting. Sample splitting prevents overfitting; because in the case of overfitting, endogeneity of the treatment may not be fully removed. We partition the sample indices into two sets $\mathcal{A}_1$ and $\mathcal{A}_2$. By default, they are approximately of size $2n/3$ and $n/3$, respectively, with $\mathcal{A}_1 = \{1, 2, \ldots, 2n/3\}$ without loss of generality. An estimator of $f(Z_i, X_i)$ for $i \in \mathcal{A}_1$ is constructed by applying some machine learning algorithm on the data $\mathcal{A}_2$ and by then predicting it on $\mathcal{A}_1$. We call these predictions $\hat{\mathbf{f}}_{\mathcal{A}_1}$. That is, the treatments $\mathbf{D}_{\mathcal{A}_2}$ are not used in this step, where $\mathbf{D}_{\mathcal{A}_2}$ contains the entries of $\mathbf{D}$ that are in $\mathcal{A}_2$. The estimated $\hat{\mathbf{f}}_{\mathcal{A}_1}$ must be of the form "hat matrix" times treatment vector, denoted by $\hat{\mathbf{f}}_{\mathcal{A}_1} = \mathbf{\Omega}\mathbf{D}_{\mathcal{A}_1}$ for some not necessarily orthogonal matrix $\mathbf{\Omega} \in \mathbb{R}^{|\mathcal{A}_1|\times|\mathcal{A}_1|}$. Such a representation is feasible for random forests if we interpret the forest as a weighted nearest neighbor method and use the induced weighting function for prediction (Lin and Jeon 2006). Indeed, for a random forest with S decision trees, the entry (i, j) of $\mathbf{\Omega}$ for $i, j \in \mathcal{A}_1$ is given by

$$\mathbf{\Omega}_{i,j} = \frac{1}{S}\sum_{s=1}^{S} \omega_j(Z_i, X_i, \theta_s) \text{ for weights } \omega_j(z, x, \theta_s) = \frac{\mathbb{1}_{\{(Z_j,X_j)\in\mathcal{R}_{l(z,x,\theta_s)}\}}}{\sum_{k\in\mathcal{A}_1}\mathbb{1}_{\{(Z_k,X_k)\in\mathcal{R}_{l(z,x,\theta_s)}\}}},$$

where $\mathcal{R}_{l(z,x,\theta_s)}$ denotes the leaf of the sth tree that contains (z, x), and θ_s denotes a random parameter that determines how the sth tree was grown. A similar representation also holds for L_2 boosting (Bühlmann and Hothorn 2007; Bühlmann and Yu 2006) with regression trees as base learners.

In **TSCI**, three methods are implemented to compute $\mathbf{\Omega}$: random forest, L_2 boosting with regression trees as base learners, and polynomial basis expansion. The former two approaches are flexible and might capture complex functions f containing nonlinearities and interactions. This is desirable because if more variation in f is captured, which machine learning algorithms

usually achieve, then the resulting IV strength is larger. The *(generalized) IV strength* measures the ratio of the variation of the treatment and the estimated treatment error variance in $\mathcal{A}_1$ after adjusting for the violation space candidate (Guo and Bühlmann 2022). If more variation in f that is orthogonal to the violation is captured, enough variation in f remains after projecting out the violation space. Preserving variation improves the efficiency of the method and gives us more room to test for IV invalidity. When testing for invalidity, our goal is to select the smallest possible violation space that can address the violation. Testing for larger spaces requires a higher IV strength. Using a polynomial approach to approximate f can be helpful in the special situation when the part of f that is orthogonal to the violation can be captured well by polynomials. In this case, the violation space sequence selection becomes particularly simple because one only has to consider subspaces spanned by elements of the respective polynomial basis. However, this approach will not be able to capture more complex features of f such as interactions and discontinuities and, thus, might lead to considerably lower IV strength than the machine learning approaches. Moreover, a polynomial expansion cannot be employed for binary IVs.

Apart from the three approaches to estimate $\mathbf{\Omega}$ that are implemented in **TSCI**, the user may provide an individual matrix. Furthermore, sample splitting is only necessary if a machine learning method is used to estimate f and is not required for the polynomial basis approach. Below, we use sample splitting notation.

2.2. Violation space selection and basis expansion

We use a set of basis function $\vec{\mathbf{v}}$ to approximate the violation function h and another set of basis functions $\vec{\mathbf{w}}$ to approximate ϕ (more precisely, the part thereof that is not orthogonal to f, that is, $\mathcal{S} \cap \text{Span}(\mathcal{V})$) on $\mathcal{A}_1$.

We choose a violation space in a data-dependent way. Given a sequence of $Q+1$ many ideally nested sets of basis functions $\{\mathcal{V}_q\}_{0 \le q \le Q}$ such that $\mathcal{V}_q \subset (\vec{\mathbf{v}} \cup \vec{\mathbf{w}})$ for every $0 \le q \le Q$ and $\mathcal{V}_0 = \vec{\mathbf{w}}$, we estimate the treatment effect β for each candidate $\mathcal{V}_q$, $0 \le q \le Q_{\max} \le Q$, where $Q_{\max}$ denotes the last candidate for which all $q \le Q_{\max}$ candidates were considered to provide enough IV strength. Subsequently, each treatment effect estimator is tested against all its successors for significant differences. Finally, the first violation space candidate $\mathcal{V}_{q_{\text{comp}}}$ for which no significant differences can be found compared to all $\mathcal{V}_{q'}$, $q_{\text{comp}} < q' \le Q_{\max}$ is selected, where the subscript "comp" stands for comparison. That is, the smallest space is selected from where onward the hypothesis of having obtained an unbiased treatment estimate could not be rejected. This approach also tests if assumptions A2 or A3 are violated at all because the first candidate, $\mathcal{V}_0 = \vec{\mathbf{w}}$, represents having no violation of A2 or A3. Consequently, if $\mathcal{V}_0$ is not selected, violations are present because the hypothesis of having no violation was rejected. A bootstrap test is used to test if the IV strength exceeds some threshold and are consequently strong enough (Guo and Bühlmann 2022). This test can also be used in case of heteroscedastic errors. Its explicit formulation is given in Guo and Bühlmann (2022).

In finite samples, some violations might not be detected by this approach. This may happen if the remaining bias due to the violation after adjusting for the selected violation space is small enough such that the treatment estimates are not found to be significantly different. Therefore, a more conservative approach to select a suitable violation space is also implemented in **TSCI**. We refer to the former approach described above as the comparison method and the latter as the conservative method. If the IVs are strong enough, the conservative

Algorithm 1 Pseudo-code of the violation space selection

Input: Treatment effect estimates for each violation space candidate: $\widehat{\beta}_0, \ldots, \widehat{\beta}_Q$. Index of the last violation space candidate that provides enough IV strength: $Q_{\max}$.
Output: Indices of the violation space candidates selected by the comparison and conservative methods: $q_{\text{comp}}, q_{\text{cons}}$.

```
procedure VIO_SPACE_SELECTION
    for q' = 0, ..., Q_max do
        significant_difference ← FALSE
        for q = q', ..., Q_max do
            if is_significantly_different(β̂_q', β̂_q) then
                significant_difference ← TRUE
            end if
        end for
        if not significant_difference then
            q_comp ← q'
        end if
    end for
    if q_comp < Q_max then
        q_cons ← q_comp + 1
    else
        q_cons ← q_comp
    end if
    return q_comp, q_cons
end procedure
```

method does not select the first violation space candidate for which no significant differences were found, but its successor in the sequence, which makes the method more conservative. In some of the simulation scenarios considered in the simulation study performed in Guo and Bühlmann (2022), the use of the conservative method was found to lead to an improvement in terms of bias and confidence interval coverage. Algorithm 1 outlines the violation space selection procedure.

We leave the specification of a suitable basis to approximate ϕ to the user. In case the user does not have good knowledge about a suitable basis, they can omit the covariates or treat them as additional IVs that might be invalid. Alternatively, they can use the polynomial basis expansion approach implemented in `tsci_poly`, in which case the covariates will enter the treatment and outcome model only linearly in the default setting, such that no basis specification is needed.

2.3. Estimating the treatment effect

Finally, we concatenate the basis functions to approximate $\mathbf{h}_{\mathcal{A}_1}$ and the basis functions to approximate $\boldsymbol{\phi}_{\mathcal{A}_1}$ row-wise into the *violation matrix* $\mathbf{V}_{\mathcal{A}_1}$. Consequently, $\mathbf{V}_{\mathcal{A}_1}$ is understood to contain a basis of $\mathcal{S} \cap \text{Span}(\mathcal{V})$ and we denote the $\mathbf{W}_{\mathcal{A}_1}$ as the submatrix of $\mathbf{V}_{\mathcal{A}_1}$ containing the basis to approximate $\boldsymbol{\phi}_{\mathcal{A}_1}$. The treatment effect estimator is then given by projecting $\boldsymbol{\Omega}\mathbf{Y}_{\mathcal{A}_1}$ and $\hat{\mathbf{f}}_{\mathcal{A}_1} = \boldsymbol{\Omega}\mathbf{D}_{\mathcal{A}_1}$ onto the orthogonal complement of the space spanned by $\widehat{\mathbf{V}}_{\mathcal{A}_1} = \boldsymbol{\Omega}\mathbf{V}_{\mathcal{A}_1}$,

and by afterward performing ordinary least squares with the projected quantities. We denote the corresponding projection matrix by $\mathbf{P}^{\perp}_{\widehat{\mathbf{V}}_{\mathcal{A}_1}}$. The rescaling of the violation matrix $\mathbf{V}_{\mathcal{A}_1}$ with $\boldsymbol{\Omega}$ is due to rescaling $\mathbf{Y}_{\mathcal{A}_1}$ with $\boldsymbol{\Omega}$, and the latter is required because $\boldsymbol{\Omega}$ is not necessarily orthogonal. The space spanned by $\mathbf{V}_{\mathcal{A}_1}$ is an approximation of $\mathcal{T}$. Additionally, a bias correction term, which is explicitly given in Guo and Bühlmann (2022), is subtracted from this effect size estimator to correct for any remaining correlation of $\boldsymbol{\epsilon}_{\mathcal{A}_1}$ and $\mathbf{D}_{\mathcal{A}_1}$ due to endogeneity, which results in the final estimator

$$\hat{\beta} = \frac{\mathbf{Y}_{\mathcal{A}_1}^{\top} \boldsymbol{\Omega}^{\top} \mathbf{P}^{\perp}_{\widehat{\mathbf{V}}_{\mathcal{A}_1}} \boldsymbol{\Omega} \mathbf{D}_{\mathcal{A}_1}}{\mathbf{D}_{\mathcal{A}_1}^{\top} \boldsymbol{\Omega}^{\top} \mathbf{P}^{\perp}_{\widehat{\mathbf{V}}_{\mathcal{A}_1}} \boldsymbol{\Omega} \mathbf{D}_{\mathcal{A}_1}} - (\text{bias term}). \tag{3}$$

To motivate this estimator, we rewrite its first term in Equation 3 as

$$\begin{aligned} \beta \cdot \frac{\mathbf{D}_{\mathcal{A}_1}^{\top} \boldsymbol{\Omega}^{\top} \mathbf{P}^{\perp}_{\widehat{\mathbf{V}}_{\mathcal{A}_1}} \boldsymbol{\Omega} \mathbf{D}_{\mathcal{A}_1}}{\mathbf{D}_{\mathcal{A}_1}^{\top} \boldsymbol{\Omega}^{\top} \mathbf{P}^{\perp}_{\widehat{\mathbf{V}}_{\mathcal{A}_1}} \boldsymbol{\Omega} \mathbf{D}_{\mathcal{A}_1}} + \frac{\mathbf{h}_{\mathcal{A}_1}^{\top} \boldsymbol{\Omega}^{\top} \mathbf{P}^{\perp}_{\widehat{\mathbf{V}}_{\mathcal{A}_1}} \boldsymbol{\Omega} \mathbf{D}_{\mathcal{A}_1}}{\mathbf{D}_{\mathcal{A}_1}^{\top} \boldsymbol{\Omega}^{\top} \mathbf{P}^{\perp}_{\widehat{\mathbf{V}}_{\mathcal{A}_1}} \boldsymbol{\Omega} \mathbf{D}_{\mathcal{A}_1}} + \\ \frac{\boldsymbol{\phi}_{\mathcal{A}_1}^{\top} \boldsymbol{\Omega}^{\top} \mathbf{P}^{\perp}_{\widehat{\mathbf{V}}_{\mathcal{A}_1}} \boldsymbol{\Omega} \mathbf{D}_{\mathcal{A}_1}}{\mathbf{D}_{\mathcal{A}_1}^{\top} \boldsymbol{\Omega}^{\top} \mathbf{P}^{\perp}_{\widehat{\mathbf{V}}_{\mathcal{A}_1}} \boldsymbol{\Omega} \mathbf{D}_{\mathcal{A}_1}} + \frac{\boldsymbol{\epsilon}_{\mathcal{A}_1}^{\top} \boldsymbol{\Omega}^{\top} \mathbf{P}^{\perp}_{\widehat{\mathbf{V}}_{\mathcal{A}_1}} \boldsymbol{\Omega} \mathbf{D}_{\mathcal{A}_1}}{\mathbf{D}_{\mathcal{A}_1}^{\top} \boldsymbol{\Omega}^{\top} \mathbf{P}^{\perp}_{\widehat{\mathbf{V}}_{\mathcal{A}_1}} \boldsymbol{\Omega} \mathbf{D}_{\mathcal{A}_1}}. \end{aligned} \tag{4}$$

The first term in Equation 4 simplifies to the true causal effect β. The second term in Equation 4 is the bias due to having invalid IVs, but it is small if the part of $\mathbf{h}_{\mathcal{A}_1}$ that is not orthogonal to $\hat{\mathbf{f}}_{\mathcal{A}_1}$ is approximated well by the column space of the violation matrix $\mathbf{V}_{\mathcal{A}_1}$. Analogously, the third term in Equation 4 is small if the part of $\boldsymbol{\phi}_{\mathcal{A}_1}$ that is not orthogonal to $\hat{\mathbf{f}}_{\mathcal{A}_1}$ is approximated well by the column space of $\mathbf{V}_{\mathcal{A}_1}$ or more specifically by the column space of the submatrix $\mathbf{W}_{\mathcal{A}_1}$. Lastly, the bias correction term in Equation 3 corrects for an potential overfitting bias caused by the fourth term in Equation 4 because $\boldsymbol{\epsilon}_{\mathcal{A}_1}$ and $\mathbf{D}_{\mathcal{A}_1}$ are correlated if endogenous confounding is present. In the case of overfitting, part of this correlation might remain even after rescaling $\boldsymbol{\epsilon}_{\mathcal{A}_1}$ and $\mathbf{D}_{\mathcal{A}_1}$ by $\boldsymbol{\Omega}$ and subsequently projecting onto the orthogonal complement of $\widehat{\mathbf{V}}_{\mathcal{A}_1}$.

2.4. Inference

If $\mathbf{V}_{\mathcal{A}_1}$ is approximating the parts of $\mathbf{h}_{\mathcal{A}_1}$ and $\boldsymbol{\phi}_{\mathcal{A}_1}$ that are not orthogonal to $\hat{\mathbf{f}}_{\mathcal{A}_1}$ well and the IVs are strong enough, it can be shown that $(\hat{\beta} - \beta)/\widehat{\mathrm{SE}}(\mathbf{V}_{\mathcal{A}_1}) \xrightarrow{d} \mathcal{N}(0, 1)$ under some regularity conditions, where $\widehat{\mathrm{SE}}(\mathbf{V}_{\mathcal{A}_1})$ denotes an estimator of the standard deviation of the treatment effect estimator. The user can choose between using the estimator of the standard deviation proposed in Guo and Bühlmann (2022) or an alternative bootstrap approach (see appendix B). The bootstrap approach is motivated by its potential to provide more accurate finite-sample tests and confidence intervals, in line with classical theory and based on some empirical evidence. In either way, the asymptotic Gaussian approximation of the treatment effect estimator is used to compute confidence intervals and p values.

2.5. Counteracting the randomness of sample splitting

Splitting the sample in two parts is random and may affect the treatment effect estimator. Thus, the whole procedure is repeated multiple times. The treatment effect estimates from

Function	Description
`tsci_forest`	Uses random forests to fit the treatment model. Requires the user to specify a set of violation space candidates.
`tsci_boosting`	Uses boosting to fit the treatment model. Requires the user to specify a set of violation space candidates.
`tsci_poly`	Uses polynomial basis expansion to fit the treatment model. Does not require the user to specify a set of violation space candidates.
`tsci_secondstage`	Does not fit the treatment model but uses a user-provided hat matrix instead. Requires the user to specify a set of violation space candidates.

Table 1: **TSCI**'s four main functions for treatment effect size estimation.

the different repetitions are aggregated by the median, and the user may choose between two schemes to compute aggregated p values and confidence intervals (and standard errors in the `"DML"` scheme presented below). The first method, called `"FWER"` in **TSCI**, leads to valid p values by controlling family-wise error rates and has initially been proposed by Meinshausen, Meier, and Bühlmann (2009). The obtained p values may be conservative. The second method, called `"DML"` in **TSCI**, directly aggregates the standard error estimators from the individual repetitions and has initially been proposed by Chernozhukov *et al.* (2018).

3. Using TSCI

Subsequently, we illustrate the use of **TSCI** using an empirical dataset.

The first step in applying **TSCI** is to select a learner for the treatment model. **TSCI** offers four built-in options, and they are given in Table 1. Their most important input arguments are presented in Table 2.

Subsequently, we demonstrate the use of **TSCI** using the dataset from Card (1993). This dataset consists of $n = 3010$ observations, and the aim is to estimate the causal effect of education (`educ`) on log earnings (`lwage`) using proximity to a 4-year college (`nearc4`) as an instrument. As exogenous baseline covariates `X`, we consider experience (`exper`, `expersq`), race (`race`), and geographical information (`smsa`, `smsa1966`, `south`, `reg661`–`reg668`). The dataset is available in the R package **ivmodel** (Kang *et al.* 2021, 2023), a package that offers extensive functionalities for classical linear IV models, and has also been analyzed by Guo and Bühlmann (2022). There are missing values in the variables `fatheduc` and `motheduc` which contain information about the years of education of the subject's parent. Since we will need these variables later on, we follow Card (1993), who uses mean-imputation and dummy variables to indicate when a value was imputed.

```
R> library("TSCI")
R> library("ivmodel")
R> data("card.data", package = "ivmodel")
R> card.data$fatheduc_na <- as.numeric(is.na(card.data$fatheduc))
R> card.data$motheduc_na <- as.numeric(is.na(card.data$motheduc))
```

Arguments	Description	Default value
`Y`	Numeric vector of outcomes.	None
`D`	Vector of treatments.	None
`Z`	Matrix or data frame of instruments.	None
`X`	Matrix or data frame of covariates for the treatment model.	`NULL`
`W`	(Transformed) observations of baseline covariates `X` for the outcome model.	`X`
`vio_space`	List to specify the violation space candidates.	None
`create_nested_sequence`	Logical. If `TRUE`, the violation space candidates are defined sequentially starting with `W` and subsequently adding the next element of `vio_space`. If `FALSE`, the violation space candidates are defined as by adding `W` to the elements of `vio_space`.	`TRUE`
`sel_method`	Selection method to estimate the treatment effect. Either `"comparison"` or `"conservative"`; see Section 2.2.	`"comparison"`
`split_prop`	Proportion of observations used to fit the outcome model, a value in $(0, 1)$.	`2/3`
`sd_boot`	Logical. If `TRUE`, it determines the standard error using a bootstrap approach.	`TRUE`
`iv_threshold`	Minimal value of threshold of IV strength test.	`10`
`threshold_boot`	Logical. If `TRUE`, the threshold of the IV strength is determined using a bootstrap approach to adjust for estimation error of the IV strength, which leads to a more conservative IV strength test.	`TRUE`
`nsplits`	Number of data splits.	`10`
`mult_split_method`	Method to calculate standard errors, p values, and to construct the confidence intervals if multi-splitting is performed. Either `"FWER"` or `"DML"`; see Section 2.5.	`"FWER"`

Table 2: Most important arguments of **TSCI**'s four main functions in Table 1. Default values are shown for `tsci_forest` and `tsci_boosting`.

```
R> card.data$fatheduc[is.na(card.data$fatheduc)] <-
+    mean(card.data$fatheduc, na.rm = T)
R> card.data$motheduc[is.na(card.data$motheduc)] <-
+    mean(card.data$motheduc, na.rm = T)
R> card.data$parenteduc <- card.data$fatheduc * card.data$motheduc
```

The first step in applying **TSCI** is to choose one of the functions from Table 1. Below, we

demonstrate how to use `tsci_boosting` and afterwards `tsci_secondstage`. The functions `tsci_forest` and `tsci_poly` are structured analogously to `tsci_boosting`. The second step in applying **TSCI** is to select a suitable transformation of the baseline covariates to approximate ϕ. The matrix of transformed covariates can be passed to the input argument `W`. This matrix will be part of every violation space candidate. If `W` is not specified, the untransformed baseline covariates are added instead. Finally, a list containing different candidates to approximate h must be provided. By default, `tsci_boosting` creates a nested sequence of violation space candidates from this list by starting with `W` as the first violation space candidate and subsequently adding the next element of the provided list to the current violation matrix to create the next violation space candidate. For instance, to specify polynomials up to degree three in the instrument `Z` together with the untransformed baseline covariates `X` as the violation space candidates, it suffices to pass a list with the elements $\{\texttt{Z}, \texttt{Z}^2, \texttt{Z}^3\}$ to `tsci_boosting`. This notation holds irrespective of the number of columns of `Z`. Alternatively, the function `create_monomials` automatically creates this sequence by calling `create_monomials(Z = Z, degree = 3)`. In our demonstration of **TSCI**, we are going to consider a list of length two where the first list element corresponds to the instrument `nearc4` and the second element to the interactions of `nearc4` with the baseline covariates:

```
R> vio_space <- with(card.data, list(nearc4,
+    nearc4 * cbind(exper, expersq, black, south, smsa, smsa66, reg661,
+    reg662, reg663, reg664, reg665, reg666, reg667, reg668)))
```

There is another function `create_interactions` that can be used to create the same list; please see the comment in the above R code. To use non-nested violation space candidates, we would have to set the input parameter `create_nested_sequence` of `tsci_boosting` to `FALSE`, in which case each element of the provided list will be treated as a violation space candidate itself without adding the previous ones. After specifying the violation space candidates, we can call `tsci_boosting` to estimate the treatment effect. To lower computation time, we only perform 5 data splits (`nsplits = 5`) in parallel (`parallel = "snow"` and `ncores = 5`), limit the boosting iterations to 15 (`nrounds = 15`) with an increased learning rate of 0.6 (`eta = 0.6`), and perform 10 bootstrap repetitions to compute the standard errors (`B = 10`). However, higher values should be chosen in practice. Moreover, we fix the maximal tree depth to 6 (`max_depth = 6`) such that no hyperparameter tuning is performed and, thus, cross-validation is not necessary (`nfolds = 1`).

```
R> RNGkind("L'Ecuyer-CMRG")
R> set.seed(10)
R> Xname <- c("exper", "expersq", "black", "south", "smsa", "smsa66",
+    "reg661", "reg662", "reg663", "reg664", "reg665", "reg666",
+    "reg667", "reg668")
R> fit_boosting <- tsci_boosting(Y = card.data$lwage, D = card.data$educ,
+    Z = card.data$nearc4, X = card.data[, Xname], vio_space = vio_space,
+    nsplits = 5, nrounds = 15, eta = 0.6, max_depth = 6,
+    nfolds = 1, B = 10, parallel = "snow", ncores = 5)
```

There is a `summary` method to provide an overview of the most relevant statistics, and it has the same form for `tsci_forest`, `tsci_boosting`, `tsci_poly`, and `tsci_secondstage`.

```
R> summary(fit_boosting)

Statistics about the data splitting procedure:
Sample size A1: 2007
Sample size A2: 1003
Number of data splits: 5
Aggregation method: FWER

Statistics about the validity of the instrument(s):
       valid      invalid non_testable
           1            4            0

Treatment effect estimate of selected violation space candidate(s):
              Estimate Std_Error    2.5%   97.5% Pr(>|t|)
TSCI-Estimate  0.05864         - 0.02521 0.08595  6.8e-05
Selection method: comparison

Statistics about the treatment model:
Estimation method: L2 Gradient Tree Boosting

Statistics about the violation space selection:
   q_comp q_cons Qmax
q0      1      0    0
q1      1      1    0
q2      3      4    5
```

The first part of the above summary output provides information about the data splitting procedure. The number of data points to estimate the treatment model (`Sample size A1`) and the outcome model (`Sample size A2`) are given first. Their size can be adjusted via the input argument `split_prop`. If not specified otherwise, 10 data splits (input argument `nsplits`) are performed to counteract the randomness introduced by the data splitting, and the `"FWER"` method (input argument `mult_split_method`) is used to aggregate the estimators obtained from each data split. Next, the result of testing for invalidity of the IV is presented. In the above output, in 1 of the 5 data splits performed by `tsci_boosting`, the IVs were found to be valid, and 4 were invalid. That is, assumptions A2 or A3 were found to be violated in 4 cases. The number of data splits for which the IV strength was too weak to estimate the treatment effect for all violation space candidates apart from the space assuming no violation are listed under `non_testable`. In our example, this did not occur in any of the 5 data splits. Next, the treatment effect obtained by the selected violation space candidate(s) and the method used to obtain the hat matrix $\mathbf{\Omega}$ are displayed. By default, the bootstrap approach is used to obtain the standard errors. Finally, the number of times (out of the 5) each violation space candidate was selected by the comparison method (first column) and the conservative method (second column) are displayed alongside with the number of times each violation space candidate was found to be the largest violation space candidate for which the IVs were strong enough (third column). In cases where this violation space candidate coincides with the violation space candidate selected by the comparison method (and consequently also by the conservative method), the treatment effect estimate should be interpreted carefully.

Indeed, the bias caused by a violation of assumptions A2 or A3 might not be fully addressed because it is not possible to reliably test whether the treatment effect estimate would change significantly when using a larger violation space candidate. The user can choose whether the treatment effect should be estimated using the comparison method or the conservative method by specifying the input parameter `sel_method`. By default, the comparison method is used, as was the case for this example. To obtain a more detailed summary of the output, we can set `extended_output = TRUE`. Then, the treatment effect estimate, the estimated IV strength, and the threshold for considering IVs as being strong enough are specified for each violation space candidate. If more than one data split is performed, the treatment effect estimate obtained by the selected violation space candidates might not coincide with any of those point estimates as in each data split a different violation space candidate might be selected. We can see that this is indeed the case here.

```
R> summary(fit_boosting, extended_output = TRUE)

Statistics about the data splitting procedure:
Sample size A1: 2007
Sample size A2: 1003
Number of data splits: 5
Aggregation method: FWER

Statistics about the validity of the instrument(s):
       valid      invalid non_testable
           1            4            0

Treatment effect estimate of selected violation space candidate(s):
              Estimate Std_Error    2.5%   97.5% Pr(>|t|)
TSCI-Estimate  0.05864         - 0.02521 0.08595  6.8e-05
Selection method: comparison

Treatment effect estimates of all violation space candidates:
        Estimate Std_Error   2.5%   97.5%  Pr(>|t|)
TSCI-q0  0.06556         - 0.0301 0.09241 1.451e-05
TSCI-q1  0.06514         - 0.0296 0.09202 7.968e-05
TSCI-q2  0.05864         - 0.0252 0.08596 0.0001809

Statistics about the treatment model:
Estimation method: L2 Gradient Tree Boosting

Statistics about the violation space selection:
   q_comp q_cons Qmax
q0      1      0    0
q1      1      1    0
q2      3      4    5

Statistics about the IV strength:
   IV_Strength IV_Threshold
```

```
q0       186.4           40
q1       186.4           40
q2       161.3           40
```

Other methods to extract relevant information are `coef`, which returns the treatment effect estimate, and `cofint`, which returns the confidence interval of the treatment effect estimate at the specified confidence level.

If we have reason to believe that the untransformed baseline covariates are not able to approximate ϕ sufficiently well, we can set the function argument `W` of `tsci_boosting`. In that case, the original baseline covariates `X` are used to fit the treatment model and the transformed baseline covariates `W` are used for the outcome model. For instance, column-wise concatenations of basis splines for the individual covariates may be considered:

```
R> suppressPackageStartupMessages(library("fda"))
R> X <- card.data[, c("exper", "expersq")]
R> head(X, 4)

  exper expersq
1    16     256
2     9      81
3    16     256
4    10     100

R> nknots <- 2
R> norder <- 3
R> nbasis <- nknots + norder - 2
R> W <- matrix(ncol = NCOL(X) * nbasis, nrow = NROW(X))
R> for (j in seq_len(NCOL(X))) {
+    knots <- quantile(unique(X[, j]), seq(0, 1, length = nknots))
+    basis <- create.bspline.basis(rangeval = range(knots), breaks = knots,
+      norder = norder)
+    W[, c(((j - 1) * nbasis + 1) : (j * nbasis))] <-
+      eval.basis(X[, j], basis)
+  }
R> head(W, 4)

          [,1]      [,2]      [,3]      [,4]      [,5]       [,6]
[1,] 0.0926276 0.4234405 0.4839319 0.2663262 0.4994836 0.23419013
[2,] 0.3705104 0.4763705 0.1531191 0.7172073 0.2593473 0.02344546
[3,] 0.0926276 0.4234405 0.4839319 0.2663262 0.4994836 0.23419013
[4,] 0.3194707 0.4914934 0.1890359 0.6576627 0.3066027 0.03573458
```

Here, the variable `nknots` specifies the number of knots for the basis spline basis, the variable `norder` specifies the order of the basis splines and, thus, the variable `nbasis` specifies the number of basis functions.

Specifying an individual hat matrix

The function `tsci_secondstage` offers the flexibility to specify an individual hat matrix $\mathbf{\Omega}$. This can be advantageous if we have good knowledge about the treatment model and can thus specify a model that captures more treatment variation than a random forest or boosting approach would, or if we want to project the violation into a certain space (for example, using an additive model to avoid the need to test for interactions in the violation).

We illustrate the usage of `tsci_secondstage` using again the dataset from Card (1993). Card (1993) discusses the possibility that their initially chosen instrument, college proximity, varies by family background, which in turn might affect the career path of the child. In such a case, college proximity would not be a valid instrument because assumption A2 is violated. However, Card (1993) argues that it might still be possible to estimate the effect of education on earnings under the assumption that the proximity to college will have a larger effect on education for children from poorer family backgrounds. He postulates the following treatment and outcome models:

$$D_i = \alpha_0 Z_i + \alpha_1 B_i * Z_i + \alpha_2^\intercal X_i + \delta_i, \quad \mathsf{E}[\delta_i \mid Z_i, B_i, X_i] = 0$$
$$Y_i = \beta D_i + \alpha_3 Z_i + \alpha_4^\intercal X_i + \epsilon_i, \quad \mathsf{E}[\epsilon_i \mid Z_i, X_i] = 0$$

Here, $B_i * Z_i$ denotes the interaction between family background and proximity to college. This means the function f in Equation 1 is assumed to be a linear function of Z_i, $B_i * Z_i$ and X_i, and the violation function h in Equation 2 takes on the simple form of $h(Z_i) = \alpha_3 Z_i$. To illustrate the use of `tsci_secondstage`, we are going to explore the validity of college proximity as an instrument under these model assumptions. To do this, we use the baseline covariates and proximity of college but also its interaction with a proxy for family background to estimate f and thus to obtain $\mathbf{\Omega}$. Then, we can use `tsci_secondstage` to test for a violation of assumption A2 by specifying the violation space candidates to be $\{\emptyset, \mathtt{Z}\}$. If α_1 is large enough, `tsci_secondstage` will choose $\mathbf{Z}$ as the violation space when α_3 is nonzero. Following Card (1993), we use the predicted education level in the absence of a nearby college based on race, geographic information in 1966, and family composition information at age 14 as our proxy for family background. In particular, we fit a linear model using only the subset of the dataset for which `nearc4` $= 0$. We refer to Card (1993) for more details regarding the rationale behind this approach.

```
R> card.data0 <- subset(card.data, nearc4 == 0)
R> fit_college_absence <-
+    lm(educ ~ reg661 + reg662 + reg663 + reg664 + reg665 + reg666 +
+    reg667 + reg668 + smsa66 + age + black + momdad14 + sinmom14 + step14 +
+    fatheduc + fatheduc_na + motheduc + motheduc_na + parenteduc,
+    data = card.data0)
R> card.data$family_background  <-
+    predict(fit_college_absence, newdata = card.data)
R>   Z <- with(card.data, cbind(nearc4, nearc4 * family_background))
R> vio_space <- list(card.data$nearc4)
```

Since, by assumption, f is a linear function, we use for $\mathbf{\Omega}$ the projection matrix from an ordinary least squares fit when regressing the variable education `D` on the `Z`, its interaction with `B`, and the covariates `X` including an intercept.

```
R> Xname <- c("exper", "expersq", "black", "south", "smsa", "smsa66",
+    "reg661", "reg662", "reg663", "reg664", "reg665", "reg666", "reg667",
+    "reg668", "fatheduc", "fatheduc_na", "motheduc", "motheduc_na",
+    "parenteduc", "momdad14", "sinmom14", "step14")
R> X <- card.data[, Xname]
R> A <- cbind(1, Z, as.matrix(X))
R> omega <- A %*% chol2inv(chol(t(A) %*% A)) %*% t(A)
```

Next, we call `tsci_secondstage`, passing our hat matrix to the input parameter `weight`.

```
R> fit_secondstage <- tsci_secondstage(Y = card.data$lwage,
+    D = card.data$educ, Z = Z, W = X, vio_space = vio_space, weight = omega)
R> summary(fit_secondstage, extended_output = TRUE)

Statistics about the data splitting procedure:
Sample size: 3010
No sample splitting was performed.

Statistics about the validity of the instrument(s):
       valid      invalid non_testable
           0            0            1

Treatment effect estimate of selected violation space candidate(s):
              Estimate Std_Error    2.5%  97.5%  Pr(>|t|)
TSCI-Estimate   0.1313   0.03476 0.06312 0.1994 0.0001596
Selection method: comparison

Treatment effect estimates of all violation space candidates:
        Estimate Std_Error    2.5%  97.5%  Pr(>|t|)
TSCI-q0   0.1313   0.03476 0.06312 0.1994 0.0001596
TSCI-q1   0.1251   0.04347 0.03989 0.2103 0.0040070

Statistics about the treatment model:
Estimation method: Specified by User

Statistics about the violation space selection:
   q_comp q_cons Qmax
q0      1      1    1
q1      0      0    0

Statistics about the IV strength:
   IV_Strength IV_Threshold
q0       40.21        40.00
q1       25.24        35.35
```

As shown in the last part of the summary output, the instrumental strength was not large enough to reliably test for a violation of assumptions A2 or A3 using this treatment model specification.

4. Practical guidance

In this section, we summarize the differences between `tsci_forest`, `tsci_boosting` and `tsci_poly`. The random forest and boosting machine learning approaches may be employed if little is known about f. The polynomial approach, `tsci_poly`, may be employed if the part of f that is orthogonal to the violation is believed to be captured well by polynomials. With the machine learning approaches, more care is required to choose a violation space sequence. With the polynomial approach, the polynomial basis can be used to specify the violation space candidates.

5. Summary and discussion

The primary goal of **TSCI** is to provide a user-friendly implementation of two stage curvature identification (TSCI, Guo and Bühlmann 2022). The TSCI method fills an important gap in the instrumental variable (IV) regression literature by providing a data-driven approach to test for invalidity of IVs and providing an effect size estimator that is robust to such violations. In particular, all instruments may be invalid. In contrast to existing approaches for invalid IVs, TSCI only makes the very mild assumption that the treatment model and the IV violation are of a different functional form. Machine learning is employed to fit the treatment model, which allows us to capture complex nonlinearities and interactions. Nevertheless, **TSCI** should not be treated as a black box algorithm if machine learning is used because expert knowledge about the potential functional forms of the violation is required.

Computational details

All packages used are available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/`. The empirical results in this paper were obtained using R 4.3.1. The random forest implementation in **ranger** 0.16.0 (Wright and Ziegler 2017), the boosting implementation in **xgboost** 1.7.7.1 (Chen *et al.* 2025) and the fast matrix multiplication implementation in **Rfast** 2.1.0 (Papadakis *et al.* 2025) are used in **TSCI** 3.0.5. Additionally, we used **fda** 6.1.8 (Ramsay 2025) for generating B-spline bases in this paper. We used **MASS** 7.3.60 (Venables and Ripley 2002) to generate multivariate Gaussian data. The computational complexity to compute the hat matrix using `tsci_forest` is quadratic in the number of observations, whereas it is cubic with `tsci_boosting`. Thus, using `tsci_boosting` might be significantly slower than `tsci_forest` for large datasets. When using `tsci_forest`, results might differ on different operational systems due to the way the random forest algorithm in **ranger** is implemented even when setting the same seed.

Acknowledgments

We would like to thank Wei Yuan for providing an initial implementation of the random forest code that was developed when Wei Yuan was a research assistant at Rutgers University with ZG. We would like to thank Mengchu Zheng, currently research assistant at Rutgers University with ZG, for helpful comments. CE and PB received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation

program (grant agreement No. 786461). The research of ZG was supported in part by the NSF-DMS 1811857, 2015373 and NIH-1R01GM140463-01; ZG also acknowledges financial support for visiting the Institute of Mathematical Research (FIM) at ETH Zurich.

References

Bach P, Kurz MS, Chernozhukov V, Spindler M, Klaassen S (2024). "**DoubleML**: An Object-Oriented Implementation of Double Machine Learning in R." *Journal of Statistical Software*, **108**(3), 1–56. doi:10.18637/jss.v108.i03.

Bowden J, Davey Smith G, Burgess S (2015). "Mendelian Randomization with Invalid Instruments: Effect Estimation and Bias Detection through Egger Regression." *International Journal of Epidemiology*, **44**(2), 512–525. doi:10.1093/ije/dyv080.

Bühlmann P, Hothorn T (2007). "Boosting Algorithms: Regularization, Prediction and Model Fitting." *Statistical Science*, **22**(4), 477–505. doi:10.1214/07-STS242.

Bühlmann P, Yu B (2006). "Sparse Boosting." *Journal of Machine Learning Research*, **7**, 1001–1024.

Card D (1993). "Using Geographic Variation in College Proximity to Estimate the Return to Schooling." *Working Paper 4483*, National Bureau of Economic Research. doi:10.3386/w4483.

Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H, Chen K, Mitchell R, Cano I, Zhou T, Li M, Xie J, Lin M, Geng Y, Li Y, Yuan J (2025). ***xgboost***: *Extreme Gradient Boosting*. doi:10.32614/CRAN.package.xgboost. R package version 1.7.11.1.

Chernozhukov V, Chetverikov D, Demirer M, Duflo E, Hansen C, Newey W, Robins J (2018). "Double/Debiased Machine Learning for Treatment and Structural Parameters." *The Econometrics Journal*, **21**(1), C1–C68. doi:10.1111/ectj.12097.

Emmenegger C (2022). ***dmlalg***: *Double Machine Learning Algorithms*. doi:10.32614/CRAN.package.dmlalg. R package version 1.0.2.

Emmenegger C, Bühlmann P (2021). "Regularizing Double Machine Learning in Partially Linear Endogenous Models." *Electronic Journal of Statistics*, **15**(2), 6461–6543. doi:10.1214/21-EJS1931.

Fan Q, He K, Zhong W (2020). ***naivereg***: *Nonparametric Additive Instrumental Variable Estimator and Related IV Methods*. doi:10.32614/CRAN.package.naivereg. R package version 1.0.5.

Fan Q, Zhong W (2018). "Nonparametric Additive Instrumental Variable Estimator: A Group Shrinkage Estimation Perspective." *Journal of Business & Economic Statistics*, **36**(3), 388–399. doi:10.1080/07350015.2016.1180991.

Farbmacher H (2018). "**SIVREG**: Stata Module to Perform Adaptive Lasso with Some Invalid Instruments." Statistical Software Components, Boston College Department of Economics. URL https://EconPapers.repec.org/RePEc:boc:bocode:s458394.

Guo Z (2023). "Causal Inference with Invalid Instruments: Post-Selection Problems and a Solution Using Searching and Sampling." *Journal of the Royal Statistical Society B*, **85**(3), 959–985. doi:10.1093/jrsssb/qkad049.

Guo Z, Bühlmann P (2022). "Causal Inference with Invalid Instruments: Exploring Nonlinear Treatment Models with Machine Learning." *arXiv 2203.12808v3*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2203.12808.

Guo Z, Kang H, Cai TT, Small DS (2018). "Confidence Intervals for Causal Effects with Invalid Instruments by Using Two-Stage Hard Thresholding with Voting." *Journal of the Royal Statistical Society B*, **80**(4), 793–815. doi:10.1111/rssb.12275.

Kang H, Jiang Y, Zhao Q, Small DS (2021). "**ivmodel**: An R Package for Inference and Sensitivity Analysis of Instrumental Variables Models with One Endogenous Variable." *Observational Studies*, **7**(2), 1–24. doi:10.1353/obs.2021.0029.

Kang H, Jiang Y, Zhao Q, Small DS (2023). ***ivmodel**: Statistical Inference and Sensitivity Analysis for Instrumental Variables Model*. doi:10.32614/CRAN.package.ivmodel. R package version 1.9.1.

Koo T, Lee Y, Small DS, Guo Z (2023). "**RobustIV** and **controlfunctionIV**: Causal Inference for Linear and Nonlinear Models with Invalid Instrumental Variables." *Observational Studies*, **9**(4), 97–120. doi:10.1353/obs.2023.a906625.

Koo T, Li S, Guo Z (2022a). ***controlfunctionIV**: Control Function Methods with Possibly Invalid Instrumental Variables*. doi:10.32614/CRAN.package.controlfunctionIV. R package version 0.1.1.

Koo T, Wang Z, Guo Z (2022b). ***RobustIV**: Robust Instrumental Variable Methods in Linear Models*. doi:10.32614/CRAN.package.RobustIV. R package version 0.2.5.

Li S, Guo Z (2020). "Causal Inference for Nonlinear Outcome Models with Possibly Invalid Instrumental Variables." *arXiv 2010.09922*, arXiv.org E-Print Archive. doi:10.48550/arXiv.2010.09922.

Lin Y, Jeon Y (2006). "Random Forests and Adaptive Nearest Neighbors." *Journal of the American Statistical Association*, **101**(474), 578–590. doi:10.1198/016214505000001230.

Meinshausen N, Meier L, Bühlmann P (2009). "p-Values for High-Dimensional Regression." *Journal of the American Statistical Association*, **104**(488), 1671–1681. doi:10.1198/jasa.2009.tm08647.

Papadakis M, Tsagris M, Dimitriadis M, Fafalios S, Fasiolo M, Borboudakis G, Burkardt J, Zou C (2025). ***Rfast**: A Collection of Efficient and Extremely Fast R Functions*. doi:10.32614/CRAN.package.Rfast. R package version 2.1.5.1.

Ramsay J (2025). ***fda**: Functional Data Analysis*. doi:10.32614/CRAN.package.fda. R package version 6.3.0.

R Core Team (2025). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. doi:10.32614/R.manuals. URL https://www.R-project.org/.

StataCorp (2025). *Stata Statistical Software: Release 19*. College Station.

Tchetgen Tchetgen E, Sun B, Walter S (2021). "The GENIUS Approach to Robust Mendelian Randomization Inference." *Statistical Science*, **36**(3), 443–464. doi:10.1214/20-STS802.

Theil H (1953a). "Estimation and Simultaneous Correlation in Complete Equation Systems." Central Planning Bureau, The Hague. doi:10.1007/978-94-011-2546-8_6. Mimeographed memorandum.

Theil H (1953b). "Repeated Least-Squares Applied to Complete Equation Systems." Central Planning Bureau, The Hague. Mimeographed memorandum.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York. doi:10.1007/978-0-387-21706-2.

Windmeijer F, Liang X (2021). ***CIIV***. R package version 1.0.0, URL https://github.com/xlbristol/CIIV.

Windmeijer F, Liang X, Hartwig FP, Bowden J (2021). "The Confidence Interval Method for Selecting Valid Instrumental Variables." *Journal of the Royal Statistical Society B*, **83**(4), 752–776. doi:10.1111/rssb.12449.

Wright MN, Ziegler A (2017). "**ranger**: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R." *Journal of Statistical Software*, **77**(1), 1–17. doi:10.18637/jss.v077.i01.

A. Encoding IV violation in h

Subsequently, we argue that if the IV Z_i is invalid, then $h \not\equiv 0$. Our argument illustrates that if g does not depend on Z_i, but Z_i is associated with the unmeasured confounders, this can again be represented as a model where g does depend on the IV. If g depends on Z_i, in which case the assumptions A2 or A3 are violated, the claim follows. Thus, let us consider an IV Z_i that violates A2. Consequently, we have $\mathsf{COV}(f(Z_i, X_i), \epsilon_i) \neq 0$. Let us define $G(Z_i) = \mathsf{E}[\epsilon_i \mid Z_i]$. Due to the exogeneity of X_i, we have $\mathsf{COV}(f(Z_i, X_i), \epsilon_i - G(Z_i)) = 0$ and consequently $\mathsf{COV}(f(Z_i), G(Z_i)) \neq 0$. If we redefine ϵ_i as $\epsilon_i - G(Z_i)$, we have again an error that is not correlated with the instrument, but a g that depends on Z_i, namely G.

B. Bootstrap approach for standard deviation

To simplify notation, we define

$$\mathbf{M}(\mathbf{V}_{\mathcal{A}_1}) = \mathbf{\Omega}^\top \mathbf{P}^{\perp}_{\widehat{\mathbf{V}}_{\mathcal{A}_1}} \mathbf{\Omega}. \tag{5}$$

The effect size estimator in Equation 3 can then be decomposed into

$$\hat{\beta} - \beta = b(\mathbf{V}_{\mathcal{A}_1}) + \frac{\mathbf{D}_{\mathcal{A}_1}^\top \mathbf{M}(\mathbf{V}_{\mathcal{A}_1})\epsilon}{\mathbf{D}_{\mathcal{A}_1}^\top \mathbf{M}(\mathbf{V}_{\mathcal{A}_1})\mathbf{D}_{\mathcal{A}_1}} - \frac{\sum_{i=1}^{n_1}[\mathbf{M}(\mathbf{V}_{\mathcal{A}_1})]_{ii}\hat{\boldsymbol{\delta}}(\mathbf{V}_{\mathcal{A}_1})_i \hat{\boldsymbol{\epsilon}}(\mathbf{V}_{\mathcal{A}_1})_i}{\mathbf{D}_{\mathcal{A}_1}^\top \mathbf{M}(\mathbf{V}_{\mathcal{A}_1})\mathbf{D}_{\mathcal{A}_1}}, \tag{6}$$

where $b(\mathbf{V}_{\mathcal{A}_1})$ is the sum of the second and third term in Equation 4 and may be small if a suitable candidate for the violation space is chosen, and the third term in Equation 6 is the explicit form of the bias correction term in Equation 3, where $\hat{\boldsymbol{\delta}}(\mathbf{V}_{\mathcal{A}_1}) = \mathbf{D}_{\mathcal{A}_1} - \widehat{f}$ and

$$\hat{\boldsymbol{\epsilon}}(\mathbf{V}_{\mathcal{A}_1}) = \mathbf{P}^{\perp}_{\widehat{\mathbf{V}}_{\mathcal{A}_1}, \widehat{\mathbf{W}}_{\mathcal{A}_1}} \left(Y - D \frac{\mathbf{Y}_{\mathcal{A}_1}^\top \mathbf{M}(\mathbf{V}_{\mathcal{A}_1})\mathbf{D}_{\mathcal{A}_1}}{\mathbf{D}_{\mathcal{A}_1}^\top \mathbf{M}(\mathbf{V}_{\mathcal{A}_1})\mathbf{D}_{\mathcal{A}_1}} \right).$$

The additional variance introduced by the bias correction term is asymptotically negligible under some regularity conditions (Guo and Bühlmann 2022). However, in the finite sample setting, accounting for it might improve the accuracy of a standard error estimator. Our bootstrap approach to derive such an estimator is as follows: for $1 \leq i \leq n_1$, we center the residuals of the treatment and outcome model, $\widetilde{\delta}_i = \widehat{\delta}_i - \bar{\mu}_\delta$ with $\bar{\mu}_\delta = \frac{1}{n_1}\sum_{i=1}^{n_1} \widehat{\delta}_i$ and $\widetilde{\epsilon}_i = [\widehat{\epsilon}(V_{Q_{\max}})]_i - \bar{\mu}_\epsilon$ with $\bar{\mu}_\epsilon = \frac{1}{n_1}\sum_{i=1}^{n_1}[\widehat{\epsilon}(V_{Q_{\max}})]_i$, respectively. For some natural number L and $1 \leq l \leq L$, we then generate $\delta_i^{[l]} = U_i^{[l]} \cdot \widetilde{\delta}_i$ and $\epsilon_i^{[l]} = U_i^{[l]} \cdot \widetilde{\epsilon}_i$ for $1 \leq i \leq n_1$, where $\{U_i^{[l]}\}_{1 \leq i \leq n_1}$ are i.i.d. standard normal random variables. Next, we compute the bootstrap analog of the second and third term in Equation 6 according to

$$N^{(l)} = \frac{\mathbf{D}_{\mathcal{A}_1}^\top \mathbf{M}(\mathbf{V}_{\mathcal{A}_1})\epsilon^{[l]} - \sum_{i=1}^{n_1}([\mathbf{M}(\mathbf{V}_{\mathcal{A}_1})]_{ii}\delta_i^{[l]}\epsilon_i^{[l]})}{D_{\mathcal{A}_1}^\top \mathbf{M}(\mathbf{V}_{\mathcal{A}_1})D_{\mathcal{A}_1}}.$$

Then, we obtain the empirical standard error of our effect size estimator from the bootstrap statistics $\{N^{(l)}\}_{1 \leq l \leq L}$, and we denote it by $\mathrm{SE}_{\mathrm{boot}}$. Subsequently, a two-sided 95% confidence interval is given by

$$(\widehat{\beta} - 1.96 \cdot \mathrm{SE}_{\mathrm{boot}}, \widehat{\beta} + 1.96 \cdot \mathrm{SE}_{\mathrm{boot}}).$$

Affiliation:

David Carl, Corinne Emmenegger, Peter Bühlmann
Seminar for Statistics
Department of Mathematics
ETH Zürich
Rämistrasse 101
8092 Zürich, Switzerland
E-mail: dcarl@ethz.ch, emmenegger@stat.math.ethz.ch, buhlmann@stat.math.ethz.ch

Zijian Guo
Department of Statistics
Rutgers University
110 Frelinghusen Road
Piscataway, NJ 08854, United States of America
E-mail: zijguo@stat.rutgers.edu

Journal of Statistical Software https://www.jstatsoft.org/
published by the Foundation for Open Access Statistics https://www.foastat.org/
September 2025, Volume 114, Issue 7 *Submitted:* 2023-07-04
doi:10.18637/jss.v114.i07 *Accepted:* 2024-08-31