

StatsPAI: A Unified, Agent-Native Python Toolkit for Causal Inference and Applied Econometrics

Biaoyue Wang ^{1,2}¶ and Scott Rozelle ^{1,2}

¹ Rural Education Action Program, Stanford Center on China's Economy and Institutions, Stanford University, United States  ² StatsPAI Inc., United States ¶ Corresponding author

DOI: [10.21105/joss.10604](https://doi.org/10.21105/joss.10604)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: Richard Liu 

Reviewers:

- [@Carol-seven](#)
- [@FATelarico](#)
- [@MartinSpindler](#)

Submitted: 07 April 2026

Published: 03 September 2026

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

StatsPAI is an open-source Python package for causal inference and applied econometrics. It gives empirical researchers a single interface for estimating, diagnosing, comparing, and reporting models that are otherwise spread across many specialized packages or proprietary statistical environments. A single import `statspai` as `sp` reaches estimators for the main families of applied work: regression and panel models, instrumental variables, the modern difference-in-differences and regression-discontinuity toolkits, synthetic control and matching, and machine-learning estimators of heterogeneous treatment effects. It also covers the diagnostics, robustness checks, and reporting that surround them. The full catalogue of more than 1,100 registered functions across more than 80 submodules is enumerated in the package documentation rather than here.

Results from the mature estimators share a common reporting surface, so the same calls produce a summary, a figure, a LaTeX or Word table, or a citation. StatsPAI is also agent-native. Every registered function exposes a machine-readable schema, a structured description of its arguments and outputs that a program can parse directly, together with structured failure metadata. Such schemas let large language model (LLM)-driven research assistants discover estimators, choose among alternatives, and surface a method's assumptions without parsing free-form prose, the capability that most distinguishes StatsPAI from a conventional estimator library. The source code is available at <https://github.com/brycewang-stanford/StatsPAI> and archived on Zenodo ([Wang & Rozelle, 2026](#)).

Statement of Need

Applied researchers face a fragmented software landscape. Stata offers an integrated workflow, but it is proprietary and does not expose a typed, machine-readable interface for AI-assisted analysis. R provides excellent method-specific packages such as `did` ([Callaway & Sant'Anna, 2021](#)), `rdrobust` ([Calonico et al., 2014](#)), `Synth` ([Abadie et al., 2010](#)), `grf` ([Athey et al., 2019](#)), and `lme4` ([Bates et al., 2015](#)), but these packages use different APIs, object systems, and output conventions. Python has strong pieces of the causal inference ecosystem, including DoWhy for graphical causal models ([Sharma & Kiciman, 2020](#)), EconML for machine-learning treatment effect estimation ([Battocchi et al., 2019](#)), CausalML for uplift modeling ([Chen et al., 2020](#)), and DoubleML for double/debiased machine learning ([Bach et al., 2022](#)). None of these tools, however, is intended to cover the full applied-econometrics workflow from design diagnosis through estimation, robustness, and publication output.

StatsPAI addresses this gap for graduate students, applied economists, policy researchers, and data scientists who want a Python-native workflow without giving up the breadth of Stata or the methodological depth of R. Its goal is not to replace every specialized implementation, but to provide a coherent empirical workspace built on shared formula conventions, compatible

result surfaces for mature estimators, export methods where supported, citations attached to estimators, and validation metadata that make the relationship between methods, assumptions, and evidence explicit.

State of the Field

Existing Python packages are strongest when they focus on a narrower problem: DoWhy emphasizes identification, graphical assumptions, and refutation; EconML and CausalML focus on heterogeneous effects and uplift modeling; DoubleML implements orthogonal-score estimators for a well-defined family of double machine-learning designs. These packages are complementary to StatsPAI, and several of its ideas follow the same methodological literature, including double/debiased machine learning (Chernozhukov et al., 2018), causal forests (Wager & Athey, 2018), and meta-learners (Künzel et al., 2019). In the high-dimensional and double machine-learning tradition specifically, StatsPAI builds on established reference implementations rather than around them. It provides a faithful Python port of the rigorous (data-driven) Lasso and post-double-selection estimators of the hdm package (Belloni et al., 2012, 2014; Chernozhukov et al., 2016), validated for numerical agreement with hdm, and orthogonal-score estimators aligned with the DoubleML implementations in Python and R (Bach et al., 2022, 2024). Researchers can thus reproduce the high-dimensional econometrics workflow inside the same agent-addressable interface used for the rest of the package.

The general-purpose regression toolkits applied economists already rely on, such as statsmodels and linearmodels in Python or fixest in R, sit at a different layer, supplying core regression machinery that StatsPAI builds on rather than competes with. Its contribution is therefore the integration layer itself, not a single better estimator. Contributing one more estimator to each existing project would still leave users with incompatible result classes, separate diagnostic conventions, and no unified agent-facing schema. StatsPAI earns a standalone existence by being the layer that makes a heterogeneous collection of estimators (classical and machine-learning, Python-native and R/Stata-aligned) behave as one coherent, agent-addressable workspace, with broad method coverage, shared reporting, explicit citations, stable/experimental API metadata, and cross-language parity checks.

Software Design

StatsPAI is organized around method families and a registry layer. Researchers can call focused functions, such as an IV or regression-discontinuity estimator, or use higher-level dispatchers that select among variants within a design family. The registry records function names, parameters, examples, stability tiers, limitations, citations, and schema information, making the package usable both from a notebook and from external systems such as a Model Context Protocol server. In practice an assistant can query the registry for estimators valid for a detected design, invoke one, read back structured diagnostics and assumption violations, and decide the next step through typed schemas rather than free-form prompts.

The central design choice is a shared result interface, in which estimators return structured objects that store coefficients, uncertainty estimates, diagnostics, fitted values, plots, and exporter hooks in predictable locations. This lowers the switching cost between classical econometric and modern machine-learning estimators, and makes validation easier because tests can compare common fields across implementations.

The package is implemented mainly in Python on top of NumPy, SciPy, Pandas, statsmodels, scikit-learn, and linearmodels, and supports Python 3.9 and newer. Optional accelerator backends are used only where they materially change the computation: PyTorch for neural causal estimators, JAX for selected bootstrap and linear-algebra workloads, and a Rust/PyO3 kernel for high-dimensional fixed-effect and cluster-variance routines. It is distributed via PyPI under the MIT license.

These choices carry costs worth stating plainly. A shared result interface means adding or upgrading an estimator is never purely local, and the reporting and export hooks are guaranteed only for the mature estimators, which is why auxiliary helpers advertise narrower capabilities through the registry rather than claiming a uniformity they do not have. Favouring breadth also trades against depth. For any single design, a dedicated package may expose more edge-case options than StatsPAI does today. Performance is likewise not the first priority of the default install, which runs on a pure-Python NumPy/SciPy stack and reaches for the PyTorch, JAX, or Rust backends only when present, holding those accelerated paths to the same documented numerical tolerances as the fallbacks. The Rust/PyO3 kernel in particular carries a compiled-language build cost, in the form of platform-specific wheels or a Rust toolchain plus a separately maintained crate. We contain this by keeping the kernel optional, with transparent pure-Python fallback. We accept these costs deliberately — for researchers who must compare estimators, switch designs, and produce reproducible output within one project, a single coherent, agent-addressable interface outweighs the loss of per-method specialization.

Research Impact Statement

StatsPAI ships a concrete validation and community-readiness dossier built from two complementary tracks. The first is a cross-language parity harness in which StatsPAI, R, and Stata are run on the same input bytes and their numerical output compared directly. The harness checks more than sixty estimator modules against a reference R implementation, with the large majority also checked against Stata. Closed-form estimators agree to machine precision; iterative and machine-learning estimators agree within pre-registered, documented tolerances, and the few remaining convention gaps are disclosed rather than hidden. Within this harness, the high-dimensional methods reproduce the growth-convergence, institutions-and-development, and gender-wage-gap applications published with the *hdm* package (Chernozhukov et al., 2016), and the double-machine-learning estimators are checked against DoubleML (Bach et al., 2022, 2024) on shared data, an independent check against those established reference implementations. The second track ships canonical teaching datasets in *sp.datasets*. Where the original extract is redistributable it is bundled as real data, so that analyses run on the study data rather than on a stand-in: returns-to-schooling IV (Card, 1993), job-training (Dehejia & Wahba, 1999; LaLonde, 1986), RD elections (Lee, 2008), synthetic control (Abadie et al., 2010), and four worked examples from Cunningham’s *Causal Inference: The Mixtape* (Cunningham, 2021). For the Mixtape examples the point estimates — and, where a reference implementation exists, the standard errors — agree with Stata 18 and the R *did* package to the tolerances recorded in the repository’s parity tests. The one deliberate exception is the Texas prison-expansion synthetic control, where the Mixtape’s recipe places lagged outcomes among the predictors and leaves the nested predictor- and donor-weight problem non-convex. Stata’s *synth* and StatsPAI therefore settle on different local optima with different donor weights, and the estimated effect agrees only to about three percent. StatsPAI attains the lower pre-treatment fit error there, so neither answer is wrong on its own objective, and the parity test records the disagreement instead of asserting a tolerance the design cannot support. The remaining entries, including the multi-period difference-in-differences panel (Callaway & Sant’Anna, 2021), are deterministic simulated replicas calibrated so that the canonical estimator recovers values near the published result; for those, exact numerical replication is deliberately not claimed. The suite also includes a 1000-replication coverage run for representative OLS, difference-in-differences, and strong-instrument IV designs, with empirical coverage close to the nominal 95 percent level. A reviewer-facing validation dossier and reviewer guide ship with the repository.

The near-term impact is a more reproducible workflow for applied policy evaluation. Sharing one interface, researchers can compare estimators on the same data, export tables with the same metadata, and record the citations and assumptions attached to each analysis. StatsPAI is currently being used in an ongoing working paper connected to the Rural Education Action Program at Stanford University, a within-person panel study of family screen-time contagion

across six survey waves in China, which relies on the package for its panel and within-person estimation. The impact claim rests on three things a reviewer can check directly: active use in an ongoing working paper, public distribution on PyPI, and the reproducible validation materials and worked examples bundled with the repository.

AI Usage Disclosure

Generative AI tools, including Claude Code and OpenAI ChatGPT/Codex, were used for code-generation assistance, refactoring suggestions, test scaffolding, documentation drafting, and manuscript copy-editing. Exact model identifiers were not retained for all exploratory sessions. Human authors made the core design decisions; reviewed, edited, and checked AI-assisted code and prose; and checked citations and software claims against repository evidence. The authors will not use generative AI to produce substantive responses to journal editors or reviewers. All authors take responsibility for the correctness, originality, licensing, and compliance of the package and this paper.

Author Contributions

Biaoyue Wang conceived and designed the package, implemented the estimators, registry, schema layer, and result objects, wrote the documentation, tests, and validation suites, and led the drafting of this paper. **Scott Rozelle** provided guidance on the package's design direction and target research workflows, and contributed to the writing, review, and revision of this paper. Both authors reviewed and approved the final manuscript.

Acknowledgements

The authors thank the Stanford Rural Education Action Program (REAP) research community and the CoPaper.AI team for feedback on early workflows. StatsPAI Inc. is the legal entity associated with the project, and CoPaper.AI is a commercial downstream product that may call the MIT-licensed StatsPAI package; the StatsPAI package itself is permanently open source under the MIT license. The authors are also grateful to the developers of NumPy, SciPy, Pandas, statsmodels, scikit-learn, linearmodels, PyTorch, JAX, and the broader open-source scientific Python ecosystem that StatsPAI builds upon.

References

- Abadie, A., Diamond, A., & Hainmueller, J. (2010). Synthetic control methods for comparative case studies: Estimating the effect of California's tobacco control program. *Journal of the American Statistical Association*, 105(490), 493–505. <https://doi.org/10.1198/jasa.2009.ap08746>
- Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized random forests. *The Annals of Statistics*, 47(2), 1148–1178. <https://doi.org/10.1214/18-aos1709>
- Bach, P., Chernozhukov, V., Kurz, M. S., & Spindler, M. (2022). DoubleML – an object-oriented implementation of double machine learning in Python. *Journal of Machine Learning Research*, 23(53), 1–6. <https://jmlr.org/papers/v23/21-0862.html>
- Bach, P., Kurz, M. S., 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. <https://doi.org/10.18637/jss.v108.i03>
- Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1), 1–48. <https://doi.org/10.18637/jss>

v067.i01

- Battocchi, K., Dillon, E., Hei, M., Lewis, G., Oka, P., Oprescu, M., & Syrgkanis, V. (2019). *EconML: A Python package for ML-based heterogeneous treatment effects estimation*. <https://github.com/py-why/EconML>
- Belloni, A., Chen, D., Chernozhukov, V., & Hansen, C. (2012). Sparse models and methods for optimal instruments with an application to eminent domain. *Econometrica*, 80(6), 2369–2429. <https://doi.org/10.3982/ECTA9626>
- Belloni, A., Chernozhukov, V., & Hansen, C. (2014). Inference on treatment effects after selection among high-dimensional controls. *The Review of Economic Studies*, 81(2), 608–650. <https://doi.org/10.1093/restud/rdt044>
- Callaway, B., & Sant’Anna, P. H. C. (2021). Difference-in-differences with multiple time periods. *Journal of Econometrics*, 225(2), 200–230. <https://doi.org/10.1016/j.jeconom.2020.12.001>
- Calonico, S., Cattaneo, M. D., & Titiunik, R. (2014). Robust nonparametric confidence intervals for regression-discontinuity designs. *Econometrica*, 82(6), 2295–2326. <https://doi.org/10.3982/ECTA11757>
- Card, D. (1993). *Using geographic variation in college proximity to estimate the return to schooling* (NBER Working Paper No. 4483). National Bureau of Economic Research. <https://doi.org/10.3386/w4483>
- Chen, H., Harinen, T., Lee, J.-Y., Yung, M., & Zhao, Z. (2020). *CausalML: Python package for causal machine learning*. <https://doi.org/10.48550/arXiv.2002.11631>
- 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. <https://doi.org/10.1111/ectj.12097>
- Chernozhukov, V., Hansen, C., & Spindler, M. (2016). hdm: High-dimensional metrics. *The R Journal*, 8(2), 185–199. <https://doi.org/10.32614/RJ-2016-040>
- Cunningham, S. (2021). *Causal inference: The mixtape*. Yale University Press. <https://doi.org/10.2307/j.ctv1c29t27>
- Dehejia, R. H., & Wahba, S. (1999). Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. *Journal of the American Statistical Association*, 94(448), 1053–1062. <https://doi.org/10.1080/01621459.1999.10473858>
- Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10), 4156–4165. <https://doi.org/10.1073/pnas.1804597116>
- LaLonde, R. J. (1986). Evaluating the econometric evaluations of training programs with experimental data. *American Economic Review*, 76(4), 604–620.
- Lee, D. S. (2008). Randomized experiments from non-random selection in U.S. House elections. *Journal of Econometrics*, 142(2), 675–697. <https://doi.org/10.1016/j.jeconom.2007.05.004>
- Sharma, A., & Kiciman, E. (2020). *DoWhy: An end-to-end library for causal inference*. <https://doi.org/10.48550/arXiv.2011.04216>
- Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523), 1228–1242. <https://doi.org/10.1080/01621459.2017.1319839>
- Wang, B., & Rozelle, S. (2026). *StatsPAI: A unified, agent-native Python toolkit for causal inference and applied econometrics* (Version 1.23.0). <https://doi.org/10.5281/zenodo.22085759>