README

Replication package for "Bayesian Estimation of a Spatial Dynamic Panel Data Probit Model"

Xuan Xiao, Xingbai Xu, and Dongxu Li

Manuscript: MS3789, *The Econometrics Journal*
Contact: Xuan Xiao, School of Data Science, The Chinese University of Hong Kong, Shenzhen; xiaoxuan@cuhk.edu.cn
Package date: November 5, 2026

Overview

This package provides the code for the simulations, diagnostics, and empirical analyses reported in the paper and online appendix, together with the data-cleaning code. Three MATLAB entry points run these analyses; the instructions below give the commands and run order. The numerical outputs can be arranged into tables following the paper's panel order and display precision.

All reported 500-replication Monte Carlo experiments were conducted on Xiamen University's HPC system. We also provide a serial version that allows the simulations to be run on a personal computer. Minor numerical differences may arise across hardware and MATLAB versions, but we expect the substantive conclusions to remain unchanged. The empirical analyses and industry-network simulation require restricted data, which are not included in the public package.

Data Availability and Provenance Statements

Statement about Rights

- ☑ I certify that the author(s) of the manuscript have legitimate access to and permission to use the data used in this manuscript.
- ☑ I certify that the author(s) of the manuscript have documented permission to redistribute/publish the data contained within this replication package. Appropriate permission are documented in the LICENSE.txt file (if applicable).

These statements apply to the public replication package. The replication team was given confidential access to restricted CSMAR and Wind analysis data. These data are not included in the public package and are excluded from the public license.

License for Data

The public county-network data are licensed under Creative Commons Attribution 4.0 International (CC BY 4.0), https://creativecommons.org/licenses/by/4.0/. Restricted CSMAR and Wind data are excluded from this license.

Summary of Availability

☑ Some data cannot be made publicly available.

Details on each Data Source

The M&A and financial data were obtained from the Company Research series of CSMAR; office-location information was also obtained from CSMAR. The industry classifications used to construct peer networks were obtained from Wind. Both databases were accessed through Xiamen University's institutional subscriptions. Financial and ST firms were excluded at download. CSMAR files were downloaded between July 13 and August 8, 2024; the industry classifications were downloaded from Wind on August 9, 2024. The resulting networks are held fixed over the sample period.

The CSMAR and Wind data are not included in the public package and cannot be publicly redistributed by the authors. The replication team was given confidential access to the processed analysis data for the reproducibility checks under the approved data exemption. Users can obtain access through CSMAR and Wind, either through an institutional subscription or by contacting the providers directly about registration, access conditions and fees. For CSMAR, the required sources are the Company Research financial and M&A data and office-location information; for Wind, the industry classifications described above. The cleaning code specifies the expected source filenames, fields and layouts, and documents how the analysis panels and networks are constructed. Access to a database does not mean that it supplies these author-constructed analysis files directly.

The simulations use a 760×760 county contiguity matrix covering ten states in the central United States. Two counties are connected if they share a border. Each relevant replication extracts a contiguous principal submatrix using a randomly drawn starting index and row-normalizes it, retaining isolated rows as zero. The matrix is supplied publicly in MAT and CSV formats; simulation datasets are generated by the code.

Data.Name	Data.Files	Location	Provided	Citation
CSMAR financial and M&A data	Main and Target panels; province/city networks	CSMAR	No; obtain source data from CSMAR	CSMAR (n.d.)
Wind industry classifications	Industry networks W1–W4	Wind	No; obtain source data from Wind	Wind (n.d.)
County contiguity matrix	W_760X760.mat; W_760X760.csv	code/	Yes, public	County contiguity matrix [dataset]

Dataset list

Data file	Source	Notes	Provided
W_760X760.mat	County network	Public; CSV counterpart supplied	Yes
WithWmergedata_201603_2024_03.xlsx	CSMAR	Main panel; confidential	No
targetdata_201603_2024_03.xlsx	CSMAR	Target panel; confidential	No
adj_W1_norm_201620 24.xlsx	Wind	One-digit industry network; confidential	No
adj_W2_norm_201620 24.xlsx	Wind	Two-digit industry network; confidential	No
adj_W3_norm_201620 24.xlsx	Wind	Three-digit industry network; confidential	No
adj_W4_norm_201620 24.xlsx	Wind	Four-digit industry network; confidential	No
adjW_prov_norm_201 62024.xlsx	CSMAR	Province network; confidential	No
adjW_city_norm_201 62024.xlsx	CSMAR	City network; confidential	No
network_firm_order.csv	Analysis panels	Firm-to-matrix order; confidential	No
W_industry_1digit_1616.mat	Wind-derived W1	Fixed network for S12; confidential	No

The filenames above describe the analysis inputs expected by the MATLAB code, not files supplied in the public package. Obtain the source data through CSMAR and Wind and use the cleaning code to construct the analysis panels and networks in an external data directory. Each of the eight Excel files has a CSV counterpart with the same filename stem. MATLAB reads the Excel files. The industry-network MAT contains the same matrix as adj_W1_norm_2016 2024.csv. The county CSV has no header or index column and preserves the MAT matrix entries and their order.

Identifiers, dates, and ordering. Variable definitions follow Table 3 of the paper. code/empirical_variable_dictionary.csv maps code names to paper labels; Cashflow corresponds to CurrentRatio.

The two panels each have 53,328 rows: 1,616 firms in 33 quarters, from March 31, 2016, through March 31, 2024. The first quarter supplies initial observations; estimation uses the remaining 32 quarters, or 51,712 observations. Symbol is a six-digit stock code and should be read as text to preserve leading zeros. date is stored as an Excel date in XLSX and as YYYY-MM-DD in CSV.

Rows are sorted by date and then stock code, with the same firm order in every quarter and both panels. Preserve the required rows, headers, and column order. The main panel columns are:

```
Symbol, date, Cashhold, Cashflow, firmsize, leverage,
ppeinvst, ROA, Tobinq, mktvalue, saleg, mergedata
```

The analysis code aligns lags internally. Each empirical network is $1,616 \times 1,616$, excluding its header; V1–V1616 denote positions. In network_firm_order.csv, row and Symbol identify the firm at each matrix row and column. The matrix row and column order corresponds to the firm order in the panels; preserving this alignment is necessary for the analysis.

Computational requirements

Software Requirements

The PC workflow was tested on an Apple M2 computer with 16 GB RAM, macOS 27.0.1, MATLAB R2026a, and Statistics and Machine Learning Toolbox 26.1. Simulation tests used individual replications. The serial workflow does not require Parallel Computing Toolbox or MATLAB Parallel Server; HPC execution is described separately below. Older MATLAB releases have not been tested.

The following R environment was used to validate the supplied data-cleaning code. R is not needed for the MATLAB analyses:

Software/package	Version
R	4.6.1
dplyr	1.2.1
lubridate	1.9.5
readxl	1.6.0
writexl	2.0.1
psych	2.6.9

MATLAB and its toolbox are available from https://www.mathworks.com/, and R from https://cran.r-project.org/. MATLAB helper functions are included in the package. An R package installation command is provided at the end of the run instructions.

Controlled Randomness

- ☑ Random seeds are set inside the simulation, diagnostic, and empirical programs, as detailed below.

Simulation programs use `base_seed=20260403` and construct replication-specific seeds from the saved design and task indices. The DGP-A/benchmark batch arrays and their order determine these seed indices. The Geweke diagnostic uses seed 20260519. Prior-sensitivity diagnostics use fixed-data seed 20260403 and chain seeds 9101–9109. DGP-A multi-chain diagnostics use fixed-data seed 20370404 and chain seeds 20380401–20380403. Empirical analyses also set their seeds internally.

Memory, Runtime, Storage Requirements

Summary Approximate time needed for all 500-replication simulations on a desktop, based on the serial estimates below:

☑ >14 days (estimated serial runtime)

Approximate output storage for one empirical workflow, excluding backups:

☑ 2 GB–25 GB (planning range)

Details The estimates below describe serial execution on a PC. For an initial check, a single replication can be run using the instructions below; reproducing the full Monte Carlo results requires all 500 replications.

The table reports sampling times recorded on the PC described above and linear estimates for serial execution of 500 replications. These times exclude data generation, input/output, collection, and plotting. They are not timings for the original HPC experiments.

Component	Recorded sampling time	Estimated serial total
DGP-A A1, N=50, T=30	6.73 min per replication, both methods	56.08 h for 500
Benchmark L1, N=50, T=30	7.26 min per replication, both methods	60.50 h for 500
DGP-A smallT, T=10,15,20	13.39 min per task, all three T values and both methods	111.56 h for 500
Robustness simulation	24.66 min per replication	8.6 days for 500
Industry-network simulation	IT 3.91 h; M-H 3.25 h per replication	149 days for 500, both methods
Empirical nocontrols, baseline, target	IT 3.90–4.18 h; M-H 3.32–3.41 h per case	—
Alternative networks	IT 3.94–4.34 h per case	—
Common quarterly shocks	IT 5.36–5.43 h per case	—
Empirical prior sensitivity	IT 3.98–4.04 h per case, first three cases	36.1 h for nine cases

The DGP-A A1 and benchmark L1 estimates apply only to the stated designs; larger N or T can take longer. The robustness time includes three full-panel and two half-panel chains, counting the reused full-panel chain once.

Saved diagnostic logs report 25.07 min of sampling across the six DGP-A chains (IT: 18.73 min; M-H: 6.34 min), and 45.18 min across the three profiled update steps in the nine prior-sensitivity chains. These historical computation times are provided as a guide, rather than as end-to-end timings on the PC listed above.

The complete serial workflow and the remaining individual steps have not been timed. Actual runtime varies by machine and design.

We recommend **at least 20 GiB of free disk space for one empirical workflow**, with additional space for backups or repeated runs. The nine-case empirical prior-sensitivity MAT measured 2.10 GiB, and the confidential inputs occupy about 115 MiB. Based on the tested designs, estimated raw simulation output is 2.82–25.22 MiB per 500-task batch, excluding summaries. Diagnostics that retain full chains require additional space; total peak storage has not been measured.

Description of programs/code

Program	Purpose
run_simulations.m	Serial execution of selected Monte Carlo tasks
run_diagnostics.m	Geweke, prior sensitivity, multi-chain diagnostics, and trace redraw
run_empirical.m	Empirical cases, APEs, figures, descriptive statistics, and text checks
code/collect_simulation_results.m	Common simulation collector; detects the family and batch from saved results
code/prepare_empirical_data.R	Builds analysis panels and six networks from authorized raw inputs

All analysis files are directly in code/; shared sampler and generator functions are supplied there. Complete run and collection commands are given below. The optional HPC/ directory contains the separate cluster deployment files.

License for Code

The public code and documentation are licensed under CC BY 4.0. The license notice and its scope are provided in LICENSE.txt at the public package root. Restricted commercial data and derived analysis inputs are excluded.

Instructions to Replicators

The steps below give the run order, required batch and network settings, and output locations.

Details

Setup

Extract the public package and keep any authorized restricted inputs outside it. In the example below, analysis_data/ is a user-created directory, not a supplied public dataset:

```
replication/
  3-replication-package/
  analysis_data/           # authorized restricted inputs, not supplied
```

Open MATLAB in 3-replication-package and define:

```
package_root = pwd;
replication_root = fileparts(package_root);
data_dir = fullfile(replication_root,'analysis_data');
simulation_root = fullfile(replication_root,'simulation_results');
diagnostic_root = fullfile(replication_root,'diagnostic_results');
empirical_root = fullfile(replication_root,'empirical_results');
```

The entry points set the required MATLAB paths and create output directories automatically. Each run uses a separate empty result directory. The paths above keep confidential inputs and results outside the public package. Adding the whole package recursively to the MATLAB path can cause conflicts with the copies in HPC/. Optional console logging is available through diary(new_log_file) and diaryoff; log files can be saved alongside the results.

Numerical reproducibility

Matrix solves, Cholesky factorizations and floating-point summations can produce slightly different results across MATLAB releases, numerical libraries and hardware. These differences may propagate through successive MCMC updates; a change in an M-H acceptance decision can also alter the subsequent chain. Fixed seeds therefore do not guarantee identical posterior draws across computing environments. The scripts use rng(seed) without explicitly selecting a generator algorithm, so comparisons also require matching random-generator settings.

The supplied reference outputs allow users to compare posterior means, standard deviations and intervals, allowing for small numerical differences across computing environments.

Analysis data

The empirical analyses read the processed Excel files listed above from data_dir. During confidential verification, the replication team used the analysis files provided by the authors. Public users must obtain authorized inputs before running these analyses. If the processed files are already available under an appropriate authorization, cleaning is not a prerequisite.

For authorized raw inputs, code/prepare_empirical_data.R lists the 12 required Excel files in input_files and specifies their expected columns. Retain the original export layout, including descriptive rows. Pass the input directory and

a separate empty output directory as the first two arguments; do not use the input data directory for output.

Simulations and collection

Each DGP-A or benchmark batch uses the N_values and T_values saved in the indicated source file; these are the only settings to edit before each call. The commands below do not set these arrays. Batches are run sequentially, with each batch completed before the next edit and saved in a separate empty directory. Keeping A1–A3 separate, A4 together, and each benchmark combination in its own singleton batch preserves the array positions used to determine seed indices.

DGP-A A1–A4. Edit code/run_simulation_dgp_a.m before each call:

A1: N_values=[50], T_values=[30].

```
run_simulations('run_simulation_dgp_a',1:500, ...
    fullfile(simulation_root,'DGP_A','A1'));
```

A2: N_values=[50], T_values=[60].

```
run_simulations('run_simulation_dgp_a',1:500, ...
    fullfile(simulation_root,'DGP_A','A2'));
```

A3: N_values=[50], T_values=[120].

```
run_simulations('run_simulation_dgp_a',1:500, ...
    fullfile(simulation_root,'DGP_A','A3'));
```

A4: N_values=[100,200], T_values=[30,60,120].

```
run_simulations('run_simulation_dgp_a',1:500, ...
    fullfile(simulation_root,'DGP_A','A4'));
```

Correctly specified benchmark, L1–L9. Edit code/run_simulation_benchmark.m before each call. Each batch has a single N and T, so both seed indices remain 1.

L1: N_values=[50], T_values=[30].

```
run_simulations('run_simulation_benchmark',1:500, ...
    fullfile(simulation_root,'benchmark','L1'));
```

L2: N_values=[50], T_values=[60].

```
run_simulations('run_simulation_benchmark',1:500, ...
    fullfile(simulation_root,'benchmark','L2'));
```

L3: N_values=[50], T_values=[120].

```
run_simulations('run_simulation_benchmark',1:500, ...
    fullfile(simulation_root,'benchmark','L3'));
```

L4: N_values=[100], T_values=[30].

```
run_simulations('run_simulation_benchmark',1:500, ...
    fullfile(simulation_root,'benchmark','L4'));
```

L5: N_values=[100], T_values=[60].

```
run_simulations('run_simulation_benchmark',1:500, ...
    fullfile(simulation_root,'benchmark','L5'));
```

L6: N_values=[100], T_values=[120].

```
run_simulations('run_simulation_benchmark',1:500, ...
    fullfile(simulation_root,'benchmark','L6'));
```

L7: N_values=[200], T_values=[30].

```
run_simulations('run_simulation_benchmark',1:500, ...
    fullfile(simulation_root,'benchmark','L7'));
```

L8: N_values=[200], T_values=[60].

```
run_simulations('run_simulation_benchmark',1:500, ...
    fullfile(simulation_root,'benchmark','L8'));
```

L9: N_values=[200], T_values=[120].

```
run_simulations('run_simulation_benchmark',1:500, ...
    fullfile(simulation_root,'benchmark','L9'));
```

DGP-A smallT, robustness, and industry network. No array edits are needed. code/run_simulation_dgp_a_smallT.m runs $N = [50]$, $T = [10, 15, 20]$, retaining indices 1–3 for Table S13. The robustness task runs all three DGP cases and the split-panel jackknife. The industry-network task requires W_industry_1digit_1616.mat in data_dir; pass that external directory

as the fourth argument. The restricted network remains outside the public package and is used as supplied, without reconstruction or renormalization.

```
run_simulations('run_simulation_dgp_a_smallT',1:500, ...
    fullfile(simulation_root,'smallT'));
run_simulations('run_simulation_robustness',1:500, ...
    fullfile(simulation_root,'robustness'));
run_simulations('run_simulation_industry_network',1:500, ...
    fullfile(simulation_root,'industry_network'),data_dir);
```

Collect completed batches. Each main writes raw MAT files directly to its batch directory. After the required tasks finish, collect each of the 16 directories below. Collection reads saved settings, so the current arrays in the mains do not affect it. Benchmark raw files retain the largeT_sim prefix, and their summaries retain largeT_hpc and largeT_summary identifiers; the collector reads them automatically.

```
addpath(fullfile(package_root,'code'));
collect_simulation_results( ...
    fullfile(simulation_root,'DGP_A','A1'));
collect_simulation_results( ...
    fullfile(simulation_root,'DGP_A','A2'));
collect_simulation_results( ...
    fullfile(simulation_root,'DGP_A','A3'));
collect_simulation_results( ...
    fullfile(simulation_root,'DGP_A','A4'));
```

```
collect_simulation_results( ...
    fullfile(simulation_root,'benchmark','L1'));
collect_simulation_results( ...
    fullfile(simulation_root,'benchmark','L2'));
collect_simulation_results( ...
    fullfile(simulation_root,'benchmark','L3'));
collect_simulation_results( ...
    fullfile(simulation_root,'benchmark','L4'));
collect_simulation_results( ...
    fullfile(simulation_root,'benchmark','L5'));
collect_simulation_results( ...
    fullfile(simulation_root,'benchmark','L6'));
collect_simulation_results( ...
    fullfile(simulation_root,'benchmark','L7'));
collect_simulation_results( ...
    fullfile(simulation_root,'benchmark','L8'));
collect_simulation_results( ...
    fullfile(simulation_root,'benchmark','L9'));
```

```
collect_simulation_results( ...
    fullfile(simulation_root,'smallT'));
collect_simulation_results( ...
    fullfile(simulation_root,'robustness'));
collect_simulation_results( ...
    fullfile(simulation_root,'industry_network'));
```

The collector detects the family and batch, saves summary MAT and CSV files in the input directory, requires all 500 unique tasks by default, and leaves existing summary files unchanged unless overwriting is explicitly enabled. Optional arguments are (directory,require_complete,batch_id,overwrite), with defaults true, inferred batch, and false; an explicit batch ID applies only to DGP-A/benchmark.

For a one-task test, use 1 instead of 1:500, a separate empty test_dir, and collect_simulation_results(test_dir,false). Partial summaries go under partial/ and are not full Monte Carlo results. Results without an industry-network fingerprint may trigger a collector warning; the fixed input network should be used unchanged.

Single-replication reference CSVs for all 16 batches are provided under referenc e_outputs/simulations/, with a short description in REFERENCE_OUTPUT S.txt. The smallT/ reference directory matches the run-output directory of the same name. They use task 1, 20,000 MCMC iterations and a 4,000-iteration burn-in. Compare the corresponding partial-summary CSV after rerunning; runtime columns are machine-dependent. These outputs do not replace the 500-replication results or the separate diagnostic runs. The industry-network reference CSV contains aggregate simulation summaries only; rerunning that design still requires authorized access to its fixed network.

Diagnostics

The first three commands run the diagnostic chains and automatically export Tables S1–S3 and S14. Each uses a separate empty result directory. The fourth command redraws Figures S1–S2 from the completed DGP-A chains without sampling again.

```
run_diagnostics('geweke',fullfile(diagnostic_root,'geweke'));
run_diagnostics('prior_sensitivity', ...
    fullfile(diagnostic_root,'prior_sensitivity'));
run_diagnostics('dgp_a_multichain',fullfile(diagnostic_root,'dgp_a'));
run_diagnostics('dgp_a_redraw',fullfile(diagnostic_root,'dgp_a'));
```

Numerical exports are in each case's tables/ directory. Raw outputs remain under the following paths relative to diagnostic_root:

```
geweke/run/output/diagnostics/geweke_joint/
prior_sensitivity/run/output/prior_sensitivity/
dgp_a/run/output/dgp_a_mcmc_diagnostics/
```

The redraw reads saved chains without sampling. To avoid overwriting, it stops if redrawn EPS files already exist or the picture directory is nonempty. Figure S1 uses the IT and M-H *_main_trace.eps panels; Figure S2 uses the corresponding *_ystar_trace.eps panels. The four EPS files are also copied to:

```
dgp_a/pictures/
```

Optional: export from saved chains. The following commands are alternatives for generating tables from already saved results. They are not needed after the diagnostic commands above complete successfully, because those commands export the tables automatically. These export-only calls use the same result directories and stop if the table files already exist.

```
run_diagnostics('geweke_export',fullfile(diagnostic_root,'geweke'));
run_diagnostics('prior_sensitivity_export', ...
    fullfile(diagnostic_root,'prior_sensitivity'));
run_diagnostics('dgp_a_export',fullfile(diagnostic_root,'dgp_a'));
```

Empirical analyses

Each empirical case reads the analysis data from data_dir and saves results under empirical_root/CASE/. These results may contain restricted information and remain outside the public package. An empty directory is required for each case to protect existing results. Start with:

```
run_empirical('nocontrols',data_dir,empirical_root);
run_empirical('baseline',data_dir,empirical_root);
run_empirical('target',data_dir,empirical_root);
```

Alternative networks. Before running each case below, set and save the matching w_case_name and w_file_name in code/run_empirical_altW.m, using the table below. Only these two settings need to change; the supplied default is W_prov.

Case	w_case_name	w_file_name
W2	W2_industry_2digit	adj_W2_norm_20162024.xlsx
W3	W3_industry_3digit	adj_W3_norm_20162024.xlsx
W4	W4_industry_4digit	adj_W4_norm_20162024.xlsx
W_prov	W_prov	adjW_prov_norm_20162024.xlsx

Case	w_case_name	w_file_name
W_city	W_city	adjW_city_norm_20162024.xlsx

```
run_empirical('W2',data_dir,empirical_root);
run_empirical('W3',data_dir,empirical_root);
run_empirical('W4',data_dir,empirical_root);
run_empirical('W_prov',data_dir,empirical_root);
run_empirical('W_city',data_dir,empirical_root);
```

Common quarterly shocks. Before *each* call, replace the existing active assignments for w_case_name and w_file_name in code/run_empirical_tim e_common_shock.m with the matching values below and save the file. Replace these two lines directly; do not add another pair of assignments. The supplied default is shock_W2.

Case	w_case_name	w_file_name
shock_W1	W1_industry_1digit	adj_W1_norm_20162024.xlsx
shock_W2	W2_industry_2digit	adj_W2_norm_20162024.xlsx

```
run_empirical('shock_W1',data_dir,empirical_root);
run_empirical('shock_W2',data_dir,empirical_root);
```

Remaining cases. The prior-sensitivity analysis runs all nine configurations. After the baseline estimation is complete, ape computes the effects from its saved MAT file, and figure plots the resulting APE summary CSV.

```
run_empirical('prior_sensitivity',data_dir,empirical_root);
run_empirical('prior_predictive',data_dir,empirical_root);
run_empirical('ape',data_dir,empirical_root);
run_empirical('figure',data_dir,empirical_root);
```

After baseline and shock_W1, export descriptive statistics and text checks:

```
run_empirical('all',data_dir,empirical_root,'Action','summarize');
```

This command reads the analysis data and the baseline and shock_W1 summary CSVs, without sampling. It saves empirical_descriptive_statistics.csv and empirical_text_checks.csv under empirical_root/checks/. The other empirical cases are not required for this step.

Optional HPC execution

The HPC/ directory contains the submission template and simulation entries for Xiamen University's HPC system. The original submission process uses

MATLAB's parallel.cluster.HPCServer interface and submits 100 tasks at a time. This section describes the cluster workflow; the PC commands above provide the serial alternative.

Place the files in HPC/, including W_760X760.mat, together in the cluster's shared directory. Industry-network tasks additionally require W_industry_1di git_1616.mat obtained under authorization in the same restricted deployment directory; this file is excluded from the public archive. The submission template retains the original institutional settings, including the Windows and shared-drive paths. It calls DGP_A_main_hpc.m by default; a submission copy can select the entry for another simulation.

For DGP-A, HPC/DGP_A_main_hpc.m is the single entry point. Set N_ values and T_values at the top of that file using the four PC batch arrays listed above: N=[50] with T=[30], [60], or [120], followed by N=[100,200] with T=[30,60,120]. Completing all 500 tasks before changing the arrays preserves the settings seen by workers. In a submission copy of the template, use:

```
createTask(job, @DGP_A_main_hpc, 0, {sim, datadir});
```

Results are grouped by the saved arrays under <shared_output_root>/DGP_ A/: N50_T30/, N50_T60/, N50_T120/, and N100_200_T30_60_120/. The last design runs both N values and all three T values together, retaining their original indices. For example:

```
addpath(fullfile(package_root,'code'));
collect_simulation_results( ...
    fullfile(shared_output_root,'DGP_A','N50_T30'));
```

For benchmark simulations, HPC/benchmark_hpc.m is the single entry point. Before each design, save one N_values from [50,100,200] and one T_values from [30,60,120] at the top of that file. Run the nine singleton combinations separately, completing all 500 tasks for one combination before changing the shared file. In a submission copy of the unchanged template, the task call is:

```
createTask(job, @benchmark_hpc, 0, {sim, datadir});
```

For example, N=50 and T=30 writes to <shared_output_root>/benchmar k/N50_T30/. Other combinations use the corresponding N/T directory name. Each design retains n_idx=t_idx=1 and the same seed rule. For example, collect the completed N=50, T=30 batch with:

```
addpath(fullfile(package_root,'code'));
collect_simulation_results( ...
    fullfile(shared_output_root,'benchmark','N50_T30'));
```

For S13, select DGP_A_smallT_hpc in a submission copy of bootsdistcom_DGP_A.m:

```
createTask(job, @DGP_A_smallT_hpc, 0, {sim, datadir});
```

Submit task ranges 1:100, 101:200, 201:300, 301:400, and 401:500, using the same shared output root. Each task runs $N = [50]$, $T = [10, 15, 20]$, with indices 1–3. Results go to <shared_output_root>/DGP_A/smallT/. Fresh directories and nonoverlapping task ranges prevent existing results from being overwritten. After all tasks finish, collect from the public package root:

```
addpath(fullfile(package_root,'code'));
collect_simulation_results( ...
    fullfile(shared_output_root,'DGP_A','smallT'));
```

Here, shared_output_root is the absolute path to the shared results as accessed from the machine used for collection.

R dependencies for the documented cleaning code. These packages are not needed for MATLAB replication using processed analysis data. The command installs current CRAN versions, which may differ from the validated versions listed above.

```
install.packages(c("dplyr", "lubridate", "readxl", "writexl", "psych"),
                 repos = "https://cloud.r-project.org")
```

List of tables and programs

The provided code covers the following outputs, subject to the data-access distinction below:

- ☑ Numerical results reported in the text.
- ☑ Tables and figures in the paper and online appendix, when authorized restricted inputs are available.
- ☑ Selected tables and figures using the public materials alone, as identified below.

Simulation paths below are relative to simulation_root; diagnostic paths are relative to diagnostic_root; empirical paths are relative to empirical_root. Tables 4–5, S12, S15–S20, and Figure 1 require confidential inputs. The other outputs use public inputs; Table 3 is a variable dictionary.

Figure/ Table #	Program	Line Number	Output file	Note
Tables 1–2; S4–S6	run_simulations.m;collect_simulation_results.m		DGP_A/A*/DGP_A_hpc_display_table_*.csv	DGP-A A1–A4

Figure/ Table #	Program	Line Number	Output file	Note
Table 3	empirical_variable_dictionary.csv		code/empirical_variable_dictionary.csv in the public package	Variable definitions
Table 4	run_empirical.m(summarize)		checks/empirical_descriptive_statistics.csv	Empirical checks; confidential inputs
Table 5	run_empirical.m		Case *_summary.csv	nocontrols, baseline, target; IT; confidential inputs
Figure 1	run_empirical.m(ape;figure)		figure/empirical_ape_covariate_interval_it_ndraw3000.*	ape, then figure; confidential inputs
Table S1	run_diagnostics.m(geweke)		geweke/tables/geweke_joint_Table_S1.csv	Diagnostic geweke
Tables S2–S3	run_diagnostics.m(prior_sensitivity)		prior_sensitivity/tables/it_prior_Table_S2.csv and it_prior_Table_S3.csv	Diagnostic prior_sensitivity
Tables S7–S9	run_simulations.m;collect_simulation_results.m		benchmark/L*/largeT_hpc_display_table_*.csv	Correctly specified benchmark L1–L9
Tables S10–S11	run_simulations.m;collect_simulation_results.m		robustness/*_misspec_table_*.csv and *_jackknife_table_*.csv	robustness
Table S12	run_simulations.m;collect_simulation_results.m		industry_network/industryW_display_table_*.csv	industry_network; confidential inputs
Table S13	run_simulations.m;collect_simulation_results.m		smallT/DGP_A_hpc_display_table_*.csv	DGP-A smallT
Table S14	run_diagnostics.m(dgp_a_multichain)		dgp_a/tables/dgpA_Table_S14.csv	Diagnostic dgp_a_multichain
Figures S1–S2	run_diagnostics.m(dgp_a_redraw)		Four EPS panels in the DGP-A diagnostic raw-output directory	Diagnostic dgp_a_redraw
Table S15	run_empirical.m		Case *_summary.csv	nocontrols, baseline, target; M-H; confidential inputs
Tables S16–S17	run_empirical.m(ape)		ape/empirical_structural_ape_ndraw3000_summary.csv	Empirical ape; confidential inputs
Table S18	run_empirical.m		Case *_summary.csv	W2, W3, W4, W_prov, W_city; confidential inputs
Table S19	run_empirical.m		Case *_summary.csv	shock_W1, shock_W2; confidential inputs

Figure/ Table #	Program	Line Number	Output file	Note
Table S20	run_empirical.m(prior_sensitivity)		prior_sensitivity/*_summary.csv	Empirical prior_sensitivity; confidential inputs

In-text sample and sensitivity numbers are in checks/empirical_text_checks.csv. Prior-predictive summaries are in prior_predictive/*_summary.csv. Figure 1 is exported in PNG, PDF, and EPS formats.

Reading the numerical outputs

For simulations, use full_mean, full_bias, and full_rmse for Mean, Bias, and RMSE; posterior SD is full_post_sd_mean, SD Ratio is full_post_sd_mean/full_mc_sd, and coverage is full_ci_coverage_95. Parameters follow $\lambda, \gamma, \rho, \beta_1, \beta_2, \beta_3$. Table 1 uses A1 and A3; Table 2 uses A4 with N=200 and T=30,120. S4 uses A2; S5 uses A4 with N=100; S6 uses A4 with N=200 and T=60. S7 uses L1–L3, S8 uses L4–L6, and S9 uses L7–L9. S11 uses the full and split_panel_jackknife rows and the columns est_mean, bias, and rmse.

For empirical estimates, use post_mean, post_sd, q025, and q975, retaining the paper's precision and interval flags. it denotes IT and mh denotes M-H throughout the numerical outputs. S16 uses IT total APEs; S17 uses direct, indirect, and total APEs for both methods. Scale the raw lagged-peer-rate APE mean, SD, and interval endpoints by 0.10 for S16–S17. S19–S20 select λ, γ, ρ. S18 uses W2, W3, W4, W_prov, W_city in that order. S19 uses shock_W1 then shock_W2; S20 uses all nine prior cases in their original order. For controlled regressions, paper order is $\lambda, \gamma, \rho, \beta_3$ (FirmSize), β_1 (Cashhold), β_2 (CurrentRatio), then $\beta_4, \ldots, \beta_9$.

References

Shenzhen CSMAR Data Technology Co., Ltd. n.d. *China Stock Market & Accounting Research Database (CSMAR).* https://data.csmar.com/. Downloaded July 13–August 8, 2024.

Wind Information Co., Ltd. n.d. *Wind Financial Terminal.* https://www.wind.com.cn/portal/en/WFT/index.html. Accessed August 9, 2024.

County contiguity matrix for ten states in the central United States [dataset]. Supplied with this replication package as W_760X760.mat and W_760X760.csv.

Xiao, Xuan, Xingbai Xu, and Dongxu Li. *Bayesian Estimation of a Spatial Dynamic Panel Data Probit Model.* Manuscript MS3789, *The Econometrics Journal.*

Acknowledgements

We acknowledge the High Performance Computing Platform at the Key Laboratory of Econometrics (Xiamen University), Ministry of Education, for providing computational resources. We also thank Mr. Yuan Jiajun for his technical support with the high-performance computing facilities at the School of Economics, Xiamen University.