

Replication Package

Technology Transfer and Early Industrial Development:
Evidence from the Sino-Soviet Alliance

Michela Giorcelli (UCLA) Bo Li (Peking University)

April 22, 2026

Overview

This replication package contains one self-contained workflow under `synthetic_replication/`, which replicates all results using synthesized data and code. All paths are defined relative to that root.

Replication Instructions (Start-to-Finish)

Synthetic Replication

1. Open Stata in `synthetic_replication/`.
2. Edit `Master.do` and set:

```
global path "Your/Path/Here"
```

3. Run:

```
do "Master.do"
```

This will:

- set up required Stata packages (`00_setup_stata_packages.do`);
- generate all synthesized tables and figures from the files in `Synthesized Do Files/`.

Expected Runtime

Runtime depends on machine configuration and whether bootstrap-intensive scripts are executed. On a standard desktop, full synthetic replication, including bootstrap and event-study routines, is estimated to require approximately 26–30 hours.

Data Availability Statement

The empirical analysis in this paper draws on plant-level production records, personnel files, and project documentation from Chinese state-owned steel enterprises. These data are held by state organs and their authorized custodians. Under the **Data Security Law of the People's Republic of China** (adopted June 10, 2021; effective September 1, 2021), state organs that collect or use data in the exercise of their statutory duties are required to do so strictly within the scope of those duties and in accordance with legal procedures. The plant-level records used in this paper fall within these categories of legally restricted data and cannot be made publicly available.

In compliance with these requirements, this replication package provides:

1. **All analysis code** — Python data-generation scripts and Stata do-files needed to reproduce every table and figure in the paper using the synthetic datasets.
2. **Synthetic datasets** — Calibrated to reproduce the regression coefficients, significance patterns, and sample structure of the original confidential data. The procedure for generating these synthetic data is described in Section 4. The synthetic data allow all code to be executed and all outputs to be generated in the correct format.

The authors confirm that they had legitimate access to the original data for the purposes of this research, and that the synthetic datasets included in this package do not contain, reconstruct, or permit inference of any individually identifiable records from the original confidential data.

Instructions for Data Access

The paper draws on three data sources. Each is subject to independent access restrictions administered by separate Chinese government authorities. Access procedures are described below.

1. **Steel Association Reports (钢铁工业统计报告), 1949–2000.** These reports contain annual plant-level performance data for the Chinese steel industry, including output, factor inputs, and operational indicators. The data are compiled and held by the Ministry of Industry and Information Technology of the People’s Republic of China (中华人民共和国工业和信息化部). Their use for research purposes remains restricted and requires formal approval from the Ministry. Inquiries may be directed to the Ministry’s data access office; further information on government industrial data is available at <https://wap.miit.gov.cn/gxsj/index.html>.
2. **Second Industrial Survey (第二次工业普查), 1985.** This survey contains firm-level data for 1985 only. The data are owned and administered by the National Bureau of Statistics of China (国家统计局). Access for research purposes requires formal application and approval from the Bureau. Information on published industrial census and survey data is available at <https://www.stats.gov.cn/sj/tjgb/gypcgb.html>.
3. **China Industrial Enterprises Database (全部国有及规模以上非国有工业企业数据库), 1998–2013.** This database is compiled by the National Bureau of Statistics of China (国家统计局) from quarterly and annual reports submitted by enterprises to local statistical bureaus. It covers all state-owned industrial enterprises and all non-state-owned enterprises with annual sales above RMB 5 million (raised to RMB 20 million from 2011), including domestic private firms, foreign-invested enterprises, and enterprises from Hong Kong, Macao, and Taiwan. The database spans more than 40 major industries, 90 intermediate categories, and 600+ sub-industries. Statistical variables include enterprise characteristics, financial accounts, and production and sales data.

Access to this database is available through the National Bureau of Statistics upon formal application, or through universities and research institutions that hold a licensed subscription. The database is also commercially available through the CCER Special Data Platform (CCER 特供数据系统平台) operated by Sinofin; see <https://www.ccerdata.cn/home/company>.

Data-Generating Process

Because the original plant-level data are confidential, we construct synthetic datasets that are statistically calibrated to reproduce the published regression results. This section describes the overall approach and the three-stage workflow used to generate the main synthetic panel.

Approach

The core idea is to work backwards from published regression coefficients. For each outcome variable, we know the target coefficient at several key post-treatment horizons and whether each estimate should be statistically significant. We generate a synthetic panel whose structure mirrors the original data, then adjust the outcome variables iteratively until running the same Stata regressions recovers coefficients that match the published targets within a tight tolerance (maximum absolute deviation < 0.001).

This approach ensures that all Stata do-files in the package can be run without modification and produce output that matches the paper’s tables in both magnitude and significance. The generation process is fully deterministic: setting `RANDOM_SEED = 18` in each Python script ensures identical output across runs and platforms.

Stage 1 — Baseline Initialization

The first stage creates a cross-section of 304 plants assigned to three groups: Know-How transfer (KH, $n = 98$), Physical Capital transfer (PC, $n = 91$), and No Transfers (NT, $n = 115$). For each plant we draw time-invariant characteristics including geographic distances, project characteristics (Table A.3), county characteristics (Table A.4), baseline production levels, and robustness sample flags. All characteristics are drawn from group-specific distributions whose parameters are chosen to match the paper’s summary statistics (Table 2) and pre-treatment balance tables (Tables A.3–A.4). Group means for balance-test variables are set analytically so that the balance regressions directly recover the published coefficients.

Stage 2 — Panel Construction

The cross-section is expanded to a balanced plant-year panel covering 1949–2000, yielding 15,808 observations. For each plant-year cell we record the event time τ (years since Soviet transfer arrival), treatment indicators, and all time-invariant characteristics. Plant-year-specific baseline values for every outcome variable are drawn to incorporate plant and year fixed effects with variance calibrated to match the paper’s reported standard errors. Baseline outcome values at pre-treatment event times are normalized across the three treatment groups so that pre-period parallel trends hold by construction and pre-period event-study coefficients are not spuriously significant.

Stage 3 — Coefficient-Targeted Calibration

The third stage adjusts outcome variables to match published regression coefficients. The workflow for each outcome is:

1. **Set targets.** For each outcome, we specify target coefficients for the KH and PC groups at a small set of focus event-time horizons, and record whether each coefficient should

be statistically significant. For the main panel outcomes (Tables 3–5), the focus horizons are $\tau \in \{1, 5, 10, 20\}$; for the full event study (Tables A.5–A.6), the set is extended to $\tau \in \{1, 5, 10, 20, 30, 40\}$.

2. **Inject treatment effects.** Treatment effects are added to the baseline outcome for KH and PC plants at each event time. Effects at non-focus event times are obtained by linear interpolation between focus-year anchors, producing a smooth treatment path.
3. **Evaluate in Python.** After injection, we run an approximation of the Stata regression entirely in Python — using the same fixed-effects structure, sample restrictions, and clustering — to compute the implied coefficients and standard errors without invoking Stata in the loop.
4. **Update amplitudes.** We compare the Python-estimated coefficients to the targets and update the treatment effect amplitudes at focus years using an iterative optimizer (Adam), minimizing a loss that penalizes both coefficient deviations and failures to achieve the required significance pattern.
5. **Iterate until convergence.** Steps 2–4 repeat until the maximum coefficient deviation across all focus years and both treatment groups falls below 0.001. The optimizer runs for at most 800 iterations.
6. **Write final dataset.** Once converged, the calibrated outcome values are written to the output `.dta` file using the exact same noise realizations as in the final iteration, guaranteeing that Stata regressions reproduce the target coefficients exactly.

For Table 6, which uses year fixed effects only and treatment dummies rather than event-time interactions, a separate calibration routine applies the same principles with a simpler fixed-effects structure.

Package Overview

Software requirements: Python 3.9+ (`numpy`, `pandas`, `scipy`); Stata 16+ (`reghdfe`, `outreg2`).

The package is organized into two main folders:

Synthetic/	Contains all synthesized data, code, and output for the synthetic replication workflow
Manuscript/	Contains original do-files and intermediate output for reference and manuscript replication

Within these two folders, the contents are organized as follows:

Synthetic/DGP Code/	Python scripts that generate all synthetic datasets
Synthetic/Synthesized Data/	Synthetic .dta files consumed by Stata do-files
Synthetic/Synthesized Do Files/	Stata do-files that produce all tables and figures using the synthetic data
Synthetic/Synthesized Output/	Output tables (.xls, .csv) and figures (.png) produced by the synthesized do-files
Manuscript/Original Dofiles/	Original Stata do-files as used with the confidential data
Manuscript/Intermediate Output/	Pre-processed plot datasets (Datasets/), figure do-files (Dofiles/), and figure output (Results/)

The intended workflow is as follows. First, run all Python scripts in **Synthetic/DGP Code/** to populate **Synthetic/Synthesized Data/**. Then run the do-files in **Synthetic/Synthesized Do Files/** to produce the regression tables in **Synthetic/Synthesized Output/** and the intermediate plot datasets in **Manuscript/Intermediate Output/Datasets/**. Finally, run the figure do-files in **Manuscript/Intermediate Output/Dofiles/** to produce all figures in **Manuscript/Intermediate Output/Results/**. The **Manuscript/Original Dofiles/** folder is provided for reference and documents the exact specifications applied to the confidential data; these do-files are not intended to be run with the synthetic data without modification.

File Index

Step 1 — Python Scripts to Synthetic Datasets

Each script in **DGP Code/** generates one or more .dta files written to **Synthesized Data/**.

Script	Output dataset(s)	Tables / Figs.
Synthetic main plant level Data_260224.py	steel_plants_all_tables.dta, steel_plants_all_tables_260225.dta	Tables 2–6, A.3–A.7, C.1–C.2; Figs. 2–4, A.1–A.2, C.2
Synthetic 156 Project Data - Table1.py	156projects_Table1.dta, 156projects_Figure1.dta	Table 1; Fig. 1
Synthetic Other Steel Plant Data - TableA1.py	steel_plants_tableA1_panel.dta	Table A.1
Synthetic County Investment Data - TableA7.py	county_panel_tableA7.dta	Table A.7
Synthetic Plants That Received No Soviet Transfers Data - TableA8.py	table_a8_synthetic_data.dta	Table A.8
Synthetic Politician Data - TableA9.py	synthetic_data_table_a9.dta	Table A.9
Synthetic complementary firm Data - Table7.py	table7_synthetic_data.dta	Table 7
Synthetic concurrent event Data - TableA10.py	tableA10.dta	Table A.10
Synthetic county Data - TableA13A14.py	tables_a13_a14_synthetic_data.dta	Tables A.13–A.14

Script	Output dataset(s)	Tables / Figs.
Synthetic_1998-2013_Data_TableA11A12_FIXED_A12.py	tableA11_1985.dta, tableA11_1998_2013.dta, tableA12_related.dta, tableA12_not_related.dta	Tables A.11–A.12

Step 2 — Datasets and Do-files to Output

Each row maps the input dataset and do-file to the table or figure it produces. All do-files read from Synthesized Data/ and write to Synthesized Output/.

Dataset	Do-file	Output file	Target
156projects_Table1.dta	Table 1.do	Table1_PanelA.xls, Table1_PanelB.xls	Table 1
steel_plants_all_tables.dta	Table 2.do	Table2_balancing.xls	Table 2
steel_plants_all_tables.dta	Table 3.do	table3.xls	Table 3
steel_plants_all_tables.dta	Table 4.do	table4.xls	Table 4
steel_plants_all_tables.dta	Table 5.do	table5.xls	Table 5
steel_plants_all_tables.dta	Table 6.do	table6.xls	Table 6
table7_synthetic_data.dta	Table 7.do	Table7.xls	Table 7
steel_plants_tableA1_panel.dta	tableA1.do	(Results not stored)	Table A.1
steel_plants_all_tables.dta	tableA2.do	tableA2_results.xls	Table A.2
steel_plants_all_tables.dta	tableA3A4.do	TableA3A4.xls	Tables A.3–4
steel_plants_all_tables.dta	tableA5.do	tableA5.xls	Table A.5
steel_plants_all_tables.dta	tableA6.do	tableA6.xls	Table A.6
county_panel_tableA7.dta	tableA7.do	tableA7_results.csv	Table A.7
table_a8_synthetic_data.dta	tableA8.do	tableA8_results.csv	Table A.8
synthetic_data_table_a9.dta	tableA9.do	TableA9.xls	Table A.9
tableA10.dta	tableA10.do	TableA10.xls	Table A.10
tableA11_*.dta	tableA11.do	Table A.11.xls	Table A.11
tableA12_*.dta	tableA12.do	Table A.12.xls	Table A.12
tables_a13_a14_synthetic_data.dta	tableA13A14.do	tableA13_results.xls, tableA14_results.xls	Tables A.13–14
steel_plants_all_tables.dta	table_c1.do	table_c1_results.csv	Table C.1
steel_plants_all_tables.dta	table_c2.do	table_c2_output.csv	Table C.2
pre_trend_data.dta	Fig2Fig3FigA2.do	Figure2_*.png, Figure3_*.png, FigureA2_*.png	Figs. 2, 3, A.2

Dataset	Do-file	Output file	Target
event_study_wide_*.dta	Figure4.do	Figure4_*.png	Figs. 4, A.7
event_study_wide_*.dta	FigureA3-A8.do	Robust_*.png	All Robustness Figures
steel_plants_all_tables_260225.dta	figc2_data.do + figc2_plot.do	FigureC2_*.png	Fig. C.2
156projects_Figure1.dta	Figure1.py	Figure 1	Fig. 1

Intermediate Output/

The **Intermediate Output/** folder contains pre-processed plot datasets and the do-files that generate all figures. It is organized into three subfolders: **Datasets/** holds the intermediate .dta files used as inputs to the figure do-files; **Dofiles/** contains the Stata do-files that produce each figure; and **Results/** receives the final figure output. The workflow within this folder is self-contained: do-files in **Dofiles/** read from **Datasets/** and write to **Results/**.

Do-file	Input dataset(s) in Datasets/	Figure
Figure2.do	Figure2.dta	Fig. 2
Figure3.do	Figure3.dta	Fig. 3
Figure4.do	Figure4_DID_log_output.dta, Figure4_DID_log_tfpq.dta, Figure4_single_diff_log_output.dta, Figure4_single_diff_log_tfpq.dta	Fig. 4
FigureA2.do	FigureA2.dta	Fig. A.2
FigureA3.do	FigureA3_clusterfe_log_output.dta, FigureA3_sovietfe_log_output.dta, FigureA3_sun_abraham_log_output.dta	Fig. A.3
FigureA4.do	FigureA4_clusterfe_log_tfpq.dta, FigureA4_sovietfe_log_tfpq.dta, FigureA4_sun_abraham_log_tfpq.dta	Fig. A.4
FigureA5.do	FigureA5_indicators_log_output.dta, FigureA5_nomanchuria_log_output.dta, FigureA5_pre1953_log_output.dta	Fig. A.5
FigureA6.do	FigureA6_indicators_log_tfpq.dta, FigureA6_nomanchuria_log_tfpq.dta, FigureA6_pre1953_log_tfpq.dta	Fig. A.6
FigureA7.do	FigureA7_county_cl_log_output.dta, FigureA7_plant_cl_log_output.dta, FigureA7_prefecture_cl_log_output.dta	Fig. A.7
FigureA8.do	FigureA8_county_cl_log_tfpq.dta, FigureA8_plant_cl_log_tfpq.dta, FigureA8_prefecture_cl_log_tfpq.dta	Fig. A.8
FigureC1.do	FigureC1_log_output.dta, FigureC1_log_tfpq.dta	Fig. C.1

Do-file	Input dataset(s) in Datasets/	Figure
FigureC2.do	FigureC2_log_output.dta, FigureC2_log_tfpq.dta	Fig. C.2

Reproducibility Notes

- All Python scripts set `RANDOM_SEED = 18`. Results are fully deterministic given the same NumPy version.
- Stata do-files use `outreg2` in append mode. Delete any prior output file (e.g. `table3.xls`) before running a do-file to avoid duplicate rows in the output.
- The `PlotDataGeneration` code takes an extremely long time to run (approximately 6–7 hours per figure) because it performs bootstrap computations across 90 variables over 1,000 iterations.