This repository contains the replication files for "Strategic Inattention, Inflation Dynamics, and the Non-Neutrality of Money" by Hassan Afrouzi.

Replication instructions

Requirements

You need Matlab and Stata to run the replication codes for this project. The code was last successfully run in the following environments:

1. Matlab replication files were last run on Matlab R2023a for Linux on a system with 256 GB of RAM and an AMD® Ryzen threadripper 3970x 32-core processor × 64 with Ubuntu version 20.04.6 LTS
2. Stata replication files were last run on Stata 15 on a 2017 27-inch iMac with a 4.2 GHz Quad-Core Intel Core i7 processor.

Quick Start

For successful replication, please perform the following tasks in the order in which they are presented:

1. Download the replication package for Coibion, Gorodnichenko and Kumar, 2018 from https://doi.org/10.1257/aer.20151299. Rename the downloaded folder `CGK_2018` and place it inside the `./input` folder within this package. The result should be that the folder `./input/CGK_2018/` contains the `license` file for CGK (2018) and a `data` folder that contains their replication files.

2. Download the replication package for Coibion, Gorodnichenko, Kumar and Ryngaert, 2021 from https://doi.org/10.7910/DVN/IAEJDU. Unzip the file `replication.zip` inside the downloaded folder, rename the unzipped folder `CGKR_2021`, and place it in the `./input` folder within this package. The result should be that the folder `./input/CGKR_2021/` contains the `workfiles` folder of CGKR (2021) (among their other replication folders and files, but those are not used in this project).

3. Open `main.do` in Stata, and in line 29, replace the string `/path/to/strategic-inattention` with the actual path to the *folder* on your computer that includes this README file. Lines 18-23 of `main.do` show the Stata packages that you need to run this file successfully. By default these lines are commented out, but if you do not have any of these packages installed, uncomment the corresponding line. If you are not sure, you can either uncomment all of these or run the code once to see which packages, if any, are missing and repeat the process. In the end, confirm that the `main.do` file was run successfully without any errors. The folder `./out` should now contain all the tables and figures generated by Stata.

4. Open `main.m` in Matlab. For simple replication of all figures and tables, run this file without any changes. Once you have done so, the folder `./out` should now contain all the figures and tables as presented in the paper and the Online Appendix.

   > **NOTE** You can also modify this file to re-solve or re-calibrate any of the models presented in the paper and its appendix, but that can take a relatively long time depending on the system on which you are running this file. Because of that, the `main.m` file saves the solutions of all calibrated models in `./workingdir/`, and loads them when they are present. All of these solutions are already included in this replication package to save the user time for simple replications of figures and tables. The `./workingdir/matlab_log.txt` shows the Matlab

> log of running the `main.m` for solving and calibrating all models. If the user wishes to replicate this process for any specific robustness exercise, delete all `.mat` files from the `./workingdir/` folder, and set all (or any subset) of the switches on lines 19-26 of the `main.m` file to `Y` (they are set to `N` by default) and run the main file (if you have set `replicate.Solve_with_ARMA` to `Y`, perform the task in step 5 as described below before running the `main.m` file.) Note that this process can take hours, depending on the system on which the file is running---it took around 2 hours on a system with 246 GB of RAM and a 32-core processor with parallelized computing to solve the model along with all the robustness exercises. Please see `./workingdir/matlab_log.txt` for what to expect.

5. This step is not necessary if you are only interested in replicating figures and tables. It is only required if you would like to re-solve the model for the robustness exercise with ARMA approximation in Appendix J (i.e., if you have set the `replicate.Solve_with_ARMA` to `Y`): Download Tan, Fei and Wu, Jieran (2023)'s z-Tran package from ztran @ commit ad3ff19 and place the unzipped version of the downloaded folder in `./codes/matlab/Library/`. Rename the folder to `ztran`.

Detailed Description of the Package

The rest of this README file provides a detailed description of all data, code sources, and internal files.

Results Generated by Matlab

The file `main.m` is the primary file for replication of results generated by Matlab. The code is automated and can be customized to replicate any desired result using the switches on lines 19 to 32. To re-solve and/or re-calibrate any specific model or to replicate figures and tables, simply assign the value of `'Y'` to its corresponding switch and run the `main.m` file.

> **NOTE** To replicate all the figures and tables generated by Matlab, you do not need to re-solve any of the models. Simply run the `main.m` file as it is included in the replication package to reproduce all the figures and tables as they appear in the paper.

> **NOTE** To re-solve the robustness exercise in Appendix J (switch `replicate.Solve_with_ARMA` in `main.m`), you need to download and include the ztran @ commit ad3ff19 package in `codes/matlab/Library/`. For detailed instructions and citation, see the section on Code Sources below.

Results Generated by Stata

Similarly, the file `main.do` is the primary file for replication of results generated by Stata.

> **NOTE** This section uses the publicly available data in the online replication packages of Coibion, Gorodnichenko and Kumar, 2018 and Coibion, Gorodnichenko, Kumar and Ryngaert, 2021. Before running the `main.do` file, please follow the instructions in the Data sources section below to download and include these replication packages in the `input` folder.

Once you have downloaded and included the datasets mentioned above in the `input` folder, open `main.do` file and replace the global variables `project_dir` with the path to the folder containing this readme file in line 29. The rest of the code is automated and can be customized to replicate any desired result using the switches on lines 41 to 61. To replicate a specific table or figure, simply assign the value of 1 to its corresponding switch and run the `main.do` file. Note that the code requires several Stata packages. If you

face any errors regarding missing packages, you can install them by uncommenting lines 18 to 23 of the `main.do` file and running this file once.

Data sources

The folder `input` is where all publicly available data used in this project are or should be located. It already includes public data on New Zealand's Gross Domestic Product (`hm5.xlsx`):

> Reserve Bank of New Zealand. (2022). Gross Domestic Product (M5). Retrieved from https://www.rbnz.govt.nz/statistics/series/economic-indicators/gross-domestic-product on November 27, 2022

along with the Reserve Bank of New Zealand's terms of use and license for redistribution. Reserve Bank of New Zealand is the source and copyright owner of all the materials and data contained in the `hm5.xlsx`.

As described in Steps 1 and 2 of the Quick Start section above, for successful replication, you should also place the datasets included in the replication packages of the following articles in the `input` folder:

- Coibion, Gorodnichenko and Kumar, 2018 (CGK_2018) from https://doi.org/10.1257/aer.20151299
- Coibion, Gorodnichenko, Kumar and Ryngaert, 2021 (CGKR_2021) from https://doi.org/10.7910/DVN/IAEJDU

Code sources

The folder `codes/matlab/Library` contains the following software packages and codes that were not written by me as a part of this project. The source links and references for these codes are:

- DRIPs.m @ commit 2c87541: for solving dynamic rational inattention problems by Afrouzi and Yang (2019)
- `neldmead_bounds.m`: Nelder-Mead optimization with bounds for parameters, originally by H.P. Gavin (see preamble of the file for reference).

To re-solve the robustness exercise in Appendix J of the paper you also need to download and place Tan, Fei and Wu, Jieran (2023)'s ztran @ commit ad3ff19 package in `codes/matlab/Library/` under the name `ztran`. Citation:

- Tan, Fei and Wu, Jieran, z-Tran: MATLAB Software for Solving Dynamic Incomplete Information Models (August 22, 2023)

Folder Structure

The `main.do` and `main.m` files are sufficient for all replications once all the datasets and required codes are downloaded and included in this folder according to the instructions above. All other files in this repository are called internally by `main.m` and `main.do` files and you do not need to run them independently.

As for the general structure, the `codes` folder includes all Matlab and Stata code files that are internally used by `main.m` and `main.do` files. The `input` folder is where external data files are (or should be) located. It already contians the New Zealand macro statistics and this is where the replication packages of Coibion, Gorodnichenko and Kumar, 2018 and Coibion, Gorodnichenko, Kumar and Ryngaert, 2021 should

be placed according to the instructions in the section on Data Sources above. The `workingdir` folder contains all the internally generated files by `main.m` and `main.do` files. This means the user can potentially delete everything here and run the main files to regenerate all these files. However, this takes a significant amount of time, especially for model solutions in `.mat` files. Accordingly, some of these `.mat` files are pre-saved so the user can replicate figures and tables quickly, without having to resolve all models. Finally, the `out` folder is where all the tables and figures---as generated by the `main.m` and `main.do` files---will be stored once those codes are run.

For completeness, I have also included a short description of each single file in this replication package below. For more information, see the preamble of each file.

```
├ 📜 main.do
├ 📜 main.m
├ 📜 README.md
├ 📜 LICENSE.md
├ 📂 codes
│ ├ 📂 matlab
│ │ ├ 📂 Library
│ │ │ ├ 📂 DRIPs.m
│ │ │ └ 📜 neldmead_bounds.m
│ │ ├ 📜 arima_approx.m
│ │ ├ 📜 calib_eval.m
│ │ ├ 📜 calibrate.m
│ │ ├ 📜 figure_appendix_cap.m
│ │ ├ 📜 figure_arma_comparison.m
│ │ ├ 📜 figure_capacities.m
│ │ ├ 📜 figure_concentration_multipliers.m
│ │ ├ 📜 figure_higher_order.m
│ │ ├ 📜 figure_identification.m
│ │ ├ 📜 figure_irfs.m
│ │ ├ 📜 figure_str_comps.m
│ │ ├ 📜 figure_uncer_vs_K.m
│ │ ├ 📜 find_alpha.m
│ │ ├ 📜 find_gamma.m
│ │ ├ 📜 interpolate_irfs.m
│ │ ├ 📜 KY_conjugate.m
│ │ ├ 📜 omega_norm.m
│ │ ├ 📜 simulate.m
│ │ ├ 📜 solve_model_arima.m
│ │ ├ 📜 solve_model_int_ma.m
│ │ ├ 📜 solve_models_arima.m
│ │ ├ 📜 solve_models_int_ma.m
│ │ ├ 📜 solve.m
│ │ ├ 📜 table_appendix.m
│ │ ├ 📜 table_calibration.m
│ │ ├ 📜 table_decomposition.m
│ │ ├ 📜 table_inflation.m
│ │ ├ 📜 table_output.m
│ │ └ 📜 var.m
│ ├ 📂 stata
│ │ ├ 📜 appendix_D.do
│ │ ├ 📜 figure_binscatter_data.do
```

```
│ │ ├ figure_hist_K.do
│ │ ├ figure_nowcast_errors.do
│ │ ├ gen_Kdist.do
│ │ ├ nz_ngdp.do
│ │ ├ process_data.do
│ │ ├ table_calib_eta.do
│ │ ├ table_calib_omega.do
│ │ ├ table_K_alpha_sumstats.do
│ │ ├ table_nowcast_errors.do
│ │ └ table_uncertainty_reg.do
├ input
│ ├ Terms of use - Reserve Bank of New Zealand.pdf
│ └ hm5.xlsx
├ out
├ workingdir
│ ├ atkeson_burstein.mat
│ ├ bm_arma.mat
│ ├ bm.mat
│ ├ cf_Table_6.mat
│ ├ cf_Table_J3.mat
│ ├ cf_Table_L3.mat
│ ├ cf_Table_L5.mat
│ ├ cf_Table_M3.mat
│ ├ matlab_log.txt
│ ├ robust_beta.mat
│ ├ robust_idio.mat
└ └ robust_rho.mat
```

Description of Files

- `main.do`: Main file for running Stata-generated results.
- `main.m`: Main file for reproducing Matlab-generated results.
- `README.md`: This Readme file.
- `LICENSE.md`: MIT License for using this replication package.
- `codes/`: Folder containing all Matlab and Stata codes.
- `codes/matlab/`: Folder containing all internal Matlab code files.
- `codes/matlab/Library`: Folder containing all Matlab codes that were used in this product but were not written by me. See the section on Code Sources above for citations, descriptions, and download instructions.
- `codes/matlab/arima_approx.m`: Internal Matlab function for arima approximation of a process with a unit root.
- `codes/matlab/calib_eval.m`: Internal Matlab function for evaluating the calibration loss function between data and model moments.
- `codes/matlab/calibrate.m`: Internal Matlab function for calibration of internally calibrated parameters of the model.
- `codes/matlab/figure_appendix_cap.m`: Internal Matlab function for generating figures for capacity of information processing across different models in the appendix.

- `codes/matlab/figure_arma_comparison.m`: Internal Matlab function for generating Figure J.1 in the Online Appendix.
- `codes/matlab/figure_capacities.m`: Internal Matlab function for generating Figure A.4 in the Online Appendix.
- `codes/matlab/concentration_multipliers.m`: Internal Matlab function for generating Figure 2 in the paper.
- `codes/matlab/figure_higher_order.m`: Internal Matlab function for generating Figure A.7 in the Online Appendix.
- `codes/matlab/figure_identification.m`: Internal Matlab function for generating Figure A.3 in the Online Appendix.
- `codes/matlab/figure_irfs.m`: Internal Matlab function for generating Figure A.5 in the Online Appendix.
- `codes/matlab/figure_str_comps.m`: Internal Matlab function for generating Figure A.6 in the Online Appendix.
- `codes/matlab/figure_uncer_vs_K.m`: Internal Matlab function for generating Figure 1 in the paper.
- `codes/matlab/find_alpha.m`: Internal Matlab function that given a set of parameters returns a vector of values for the degree of strategic complementarity as a function of an array for number of competitors.
- `codes/matlab/find_gamma.m`: Internal Matlab function that externally calibrates curvature of profit function to target average strategic complementarity moment.
- `codes/matlab/interpolate_irfs.m`: Internal Matlab function for interpolating IRFs on a finer time grid.
- `codes/matlab/KY_conjugate.m`: Matlab function for solving Kalman-Yakubovich equations as a fast method to generate Figure A.7.
- `codes/matlab/omega_norm.m`: Internal Matlab function that returns a vector of normalization factors for the cost of information as described in Appendix J in Eqns J.57-9.
- `codes/matlab/simulate.m`: Internal Matlab function for simulating panel data for firms' expectations given a model solution.
- `codes/matlab/solve_model_arima.m`: Internal Matlab function for solving the model with ARIMA approximation of the state space representation for the robustness exercise presented in Appendix J.
- `codes/matlab/solve_models_arima.m`: Internal Matlab function for solving the model with the ARIMA approximation for an array of parameters (e.g., information costs, number of competitors etc.) for the robustness exercise presented in Appendix J.
- `codes/matlab/solve_model_int_ma.m`: Internal Matlab function for solving the model with Integrated MA truncation of the state space representation as described in Appendix J.
- `codes/matlab/solve_models_int_ma.m`: Internal Matlab function for solving the model with Integrated MA truncation of the state space representation for an array of parameters (e.g., information costs, number of competitors etc.) as described in Appendix J.
- `codes/matlab/solve.m`: Internal wrapper function for calling different solvers (ARIMA vs. Int. MA. truncation) to solve the model for an array of parameters.
- `codes/matlab/table_appendix.m`: Internal Matlab function for generating appendix tables for output and inflation statistics across counterfactual models.
- `codes/matlab/table_calibration.m`: Internal Matlab function for generating Table 3 in the paper.

- `codes/matlab/table_decomposition.m`: Internal Matlab function for generating Table 6 in the paper.
- `codes/matlab/table_inflation.m`: Internal Matlab function for generating Table 5 in the paper.
- `codes/matlab/table_output.m`: Internal Matlab function for generating Table 4 in the paper.
- `codes/matlab/var.m`: A one-line Matlab function for calculating the inner product of a vector with itself.
- `codes/stata/`: Folder containing all internal Stata code files.
- `codes/stata/appendix_D.do`: Internal Stata do file for replication of Figure D.1 and Table D.1 in the Online Appendix.
- `codes/stata/figure_binscatter_data.do`: Internal Stata do file for generating the binscatter data used in plotting Figure 1.
- `codes/stata/figure_hist_K.do`: Internal Stata do file for replicating Figure A.1 in the Online Appendix.
- `codes/stata/figure_nowcast_errors.do`: Internal Stata do file for replicating Figure A.2 in the Online Appendix.
- `codes/stata/gen_Kdist.do`: Internal Stata do file for constructing the shares of the number of competitors in the data and saving it in `workingdir/shares.csv` for calibrating the distribution of K in the model.
- `codes/stata/nz_ngdp.do`: Internal Stata do file for constructing the moments for calibration of the process for the nominal GDP of New Zealand.
- `codes/stata/process_data.do`: Internal Stata do file to process the data in `./input/` from CGK (2018) and CGKR (2021) to use for replication of figures and tables.
- `codes/stata/table_calib_eta.do`: Internal Stata do file for replicating Tables I.1 and L.1 in the Online Appendix.
- `codes/stata/table_calib_omega.do`: Internal Stata do file for replicating Table A.2 in the Online Appendix.
- `codes/stata/table_K_alpha_sumstats.do`: Internal Stata do file for replicating Table A.1 in the Online Appendix.
- `codes/stata/table_nowcast_errors.do`: Internal Stata do file for replicating Table 2 in the paper.
- `codes/stata/table_uncertainty_reg.do`: Internal Stata do file for replicating Table 1 in the paper.
- `input/`: Folder containing external data sources.
- `input/Terms of use - Reserve Bank of New Zealand.pdf`: Copyright ownership notice and redistribution instructions of the New Zealand GDP data in `hm5.xlsx` from the Reserve Bank of New Zealand.
- `input/hm5.xlsx`: Public data for New Zealand nominal GDP (See Data Sources).
- `out/`: Folder for storing replicated figures and tables. `main.m` and `main.do` automatically save the replicated results here.
- `workingdir/`: Folder containing all internally generated Matlab and Stata data files. All the files in this folder can be regenerated using `main.m` and `main.do` files. Existing files are pre-saved model solutions to accommodate quick replication of figures and tables.
- `workingdir/atkeson_burstein.mat`: Internally generated Matlab file that stores the calibrated solution of the model with Atkeson and Burstein preferences, presented in the Online Appendix.
- `workingdir/bm_arma.mat`: Internally generated Matlab file that stores the calibrated solution of the benchmark model solved with the ARIMA approximation of the state space representation, as

presented in the Online Appendix J.

- `workingdir/bm.mat`: Internally generated Matlab file that stores the calibrated solution of the benchmark model solved with the integrated MA truncation of the state space. This is the benchmark solution for which results are reported in the main body of the paper.
- `workingdir/cf_Table_6.mat`: Internally generated Matlab file that stores the solution of the model for the counterfactual exercise reported in Table 6.
- `workingdir/cf_Table_J3.mat`: Internally generated Matlab file that stores the solution of the model for the counterfactual exercise reported in Table J3.
- `workingdir/cf_Table_L3.mat`: Internally generated Matlab file that stores the solution of the model for the counterfactual exercise reported in Table L3.
- `workingdir/cf_Table_L5.mat`: Internally generated Matlab file that stores the solution of the model for the counterfactual exercise reported in Table L5.
- `workingdir/cf_Table_M3.mat`: Internally generated Matlab file that stores the solution of the model for the counterfactual exercise reported in Table M3.
- `workingdir/matlab_log.txt`: Internally generated diary file by Matlab for solving all and calibrating all models presented in the paper and the Online Appendix.
- `workingdir/robust_beta.mat`: Internally generated Matlab file that stores the calibrated solution of the model with an internally calibrated value for the discount factor, as presented in the Online Appendix M.
- `workingdir/robust_idio.mat`: Internally generated Matlab file that stores the calibrated solution of the model with idiosyncratic shocks, as presented in the Online Appendix M.
- `workingdir/robust_rho.mat`: Internally generated Matlab file that stores the calibrated solution of the model with the alternative value of the persistence parameter for the growth of nominal GDP, as presented in the Online Appendix M.

Contact Information

If you have any questions or encounter issues with replicating the study, please feel free to reach out via email: ha2475@columbia.edu.