Replicate the Main Results

To replicate the results, proceed in the following steps. The C++ code needs to be run on a x64 platform and requires parallel computing using OpenMP with O2 optimization. All codes are written based on relative file directories.
Add folder "matlab_functions" and all subfolders to MATLAB directory.
Add folder "data" to MATLAB directory.

1. Solve the model using code in "code_main_model"

1) Run C++ code in "1_solve_ss" to solve the steady-state equilibrium corresponding to the branch distribution fixed at time t = 0.
The input of the program are the market characteristics obtained by running "1_solve_ss\input\prepare_input.m". Copy d_ini.dat, distance.dat, Pi.dat, and Z.dat to "4_segment_level\1_solve_highest_possible_1year_netprofit\no_new_branch\input", "4_segment_level\1_solve_highest_possible_1year_netprofit\open_new_branch\input", "4_segment_level\2_solve_combination\input", and "5_country_level\input".
The output of the program is located at "1_solve_ss\computation_results".
Then, run "1_solve_ss\computation_results\read_data.m" to read all the results, which are saved as "1_solve_ss\computation_results\data.mat".
The results contain the steady-state equilibrium interest rates and wages for each market and the initial distribution of households for use in step 3) and beyond.

Remark: How to run the C++ code?
If one is using Microsoft VS, create a project and add all .cpp and .h files. In the default setting, OpenMP and O2 optimization are not opened so code is not running in parallel and optimized mode. Please check online how to enable your openMP and O2 optimization in VS.
If one is using gcc in Linux OS, first create a "compile" file that includes all .cpp and .h files. Add "-fopenmp" and "-O2" to enable parallel computing and O2 optimization. See "1_solve_ss\compile" as an example. Then run "main.o" to produce the results.

2) Run Matlab code in "2_partition" (a_initial_guess.m, b_k_medoid.m, c_segment.m in turn) to partition markets into segments.
This code conducts the partition, divides markets of each segment into group 1 and group 2 markets, and saves the results in "segment.mat".

3) Run Matlab code (make_guess.m) in "3_guess_w_r" to guess the equilibrium interest rates and wages for each market along the transition path. This is just one example to make the initial guesses. This code produces r.dat, w.dat, and pdf_ini.dat. Put the three files inside "4_segment_level\1_solve_highest_possible_1year_netprofit\no_new_branch\input", "4_segment_level\1_solve_highest_possible_1year_netprofit\open_new_branch\input", "4_segment_level\2_solve_combination\input",

and
"5_country_level\input" as future input.

4) Enter folder
"4_segment_level\1_solve_highest_possible_1year_netprofit\no_new_branch".
to run C++ code.
The input of the program is produced by steps 1) (d_ini.dat, distance.dat, Pi.dat, and Z.dat) and 3) (r.dat, w.dat, and pdf_ini.dat).
This code solves the joint profits of all the existing branches in 1986, conditional on the guesses for equilibrium interest rates and wages and that no new branches are opened after 1986.
The output of the program is located at "computation_results".
Then, run "computation_results\read_data.m" to read all the results, which are saved as "computation_results\data.mat".

5) Enter folder
"4_segment_level\1_solve_highest_possible_1year_netprofit\open_new_branch" to run C++ code.
The input of the program is produced by steps 1) (d_ini.dat, distance.dat, Pi.dat, and Z.dat) and 3) (r.dat, w.dat, and pdf_ini.dat).
This code solves the highest-possible 1-year profit for a branch in each market, conditional on the guesses for equilibrium interest rates and wages.
The output of the program is located at "computation_results".
Then, run "computation_results\read_data.m" to read all the results, which are saved as "computation_results\data.mat".
Then, run "computation_results\new_branch_loc.m" to calculate highest-possible 1-year net profit for a branch in each market. This code also sorts all locations in each segment based on their highest-possible 1-year net profits from highest to lowest. For some segments, this already works out the potential location choices. For other segments, we still need to solve the potential location choices. This code saves the input to do that, in location "computation_results\combination_input".
Copy these inputs (seg.dat, seg_nnewbank_bound.dat, seg_open.dat) to
"4_segment_level\2_solve_combination\input\combination_input".

6) Enter folder "4_segment_level\2_solve_combination" to run C++ code.
The input of the program is produced by steps 1) (d_ini.dat, distance.dat, Pi.dat, and Z.dat), 3) (r.dat, w.dat, and pdf_ini.dat), and 5) (seg.dat, seg_nnewbank_bound.dat, seg_open.dat).
This code solves the potential location choices for the rest segments by comprehensively searching for all combinations.
The output of the program is located at "computation_results".
Then, run "computation_results\new_branch_loc.m" to produce inputs in "knapsack_input" for future use.
Copy these inputs (bank86.dat, nnewbank.dat, seg.dat, seg_nnewbank_bound.dat, seg_year_newbranch.dat, seg_open.dat) to "5_country_level\input\knapsack_input".

7) Enter folder “5_country_level” to run C++ code.
The input of the program is produced by steps 1) (d_ini.dat, distance.dat, Pi.dat, and Z.dat), 3) (r.dat, w.dat, and pdf_ini.dat), and 6) (bank86.dat, nnewbank.dat, seg.dat, seg_nnewbank_bound.dat, seg_year_newbranch.dat, seg_open.dat).
This code aggregates the potential location choices of all segments into country level optimal location choices by solving a multidimensional multiple-choice knapsack problem.
The output of the program is located at “computation_results”.
Then, run “computation_results\read_data.m” to read all the results, which are saved as “computation_results\data.mat”.
The results contain the optimal country-level bank expansion choices.

8) Enter folder “6_implied_w_r” to run C++ code.
The input of the program are the market characteristics and the distance (both obtained by running “input\prepare_input.m”).
This code computes the implied equilibrium interest rates and wages for each market and statistics for each market.
Then, run “computation_results\read_data.m” to read all the results, which are saved as “6_implied_w_r\computation_results\data.mat”.

9) Compare the implied equilibrium interest rates and wages with the guesses. If they are close enough, the solution is found. Otherwise, form new guesses and return to step 3).

2. Replicate the results of the main text using code in "code_figure_table\maintext"

1) Run "Figure 1\fig1.m" to replicate Figure 1.

2) Run "Figure 2\panelA.mxd" to replicate panel A of Figure 2 and "Figure 2\panelB.mxd" to replicate panel B of Figure 2. The Polyline file for panel B is provided in folder "data\distance".
   Remark:
   To run .mxd file, download software ArcMap 10.8. After launching ArcMap 10.8, if the figure is not drawn automatically, please reset the data source of each layer. The data source is provided in folder "data".

3) Run "Table 1\employment\DID.do" to replicate column (1) of Table 1.
   Run "Table 1\income\DID.do" to replicate column (2) of Table 1.
   Run "Table 1\entrepreneur\DID.do" to replicate column (3) of Table 1.
   Run "Table 1\loan\DID.do" to replicate column (4) of Table 1.

4) Data moments reported in Table 2 are saved in "Table 2\data".
   Model moments reported in Table 2 are calculated in "Table 2\model".
   i. Run "basic_stats.m" to obtain interest rate in 1986, fraction of farmers in 1986, migration rate 1986 – 1996, out-migrant share ratio, average deposit-cash ratio, sensitivity of deposit-cash ratio to d, and fraction of households with loans.
   ii. Run "rev_simulate_main.m" to obtain the standard deviation of log revenue growth and the autocorrelation of log revenue.
   iii. Run "DID_a_matching.m", "DID_b_preparedata.m", and "DID_PSM.do" in turn to obtain the DID estimates.

5) Copy the results from "Table 2\DID_PSM.do" and paste them into "Figure 5\Figure5.m". Then use "Figure 5\Figure5.m" to draw Figure 5.

6) Run "Figure 6\num_branches.m" to replicate Figure 6.

7) Run "Table 3\panelA\fill_table.m" to obtain the results of Panel A of Table 3.
   Run "Table 3\panelB\fill_table.m" to obtain the results of Panel B of Table 3.
   Note that we read the bank expansion choices from model results saved in "code_main_model\6_implied_w_r\input".

8) Run "Figure 7\Figure_7.m" to replicate Figure 7.

9) To replicate Figure 8, follow the C++ code in "code_main_model\6_implied_w_r" to solve the model with only credit channel and with only deposit channel. In particular, we hold the locations of new branches unchanged from our main model. Then, use "Figure 8\calculation_aggregate.m" to obtain results. Run "Figure 8\Figure8.m" to replicate Figure 8. The data series are also reported in "Figure 8\Figure8.m".

10) To replicate Figure 9, 10, use “Figure 9_10\model\calculate_gini.m” to obtain the results. Run “Figure 9 10\Figure9.m” to replicate Figure 9 and run “Figure 9 10\Figure10.m” to replicate Figure 10.

11) To replicate Figure 11, use “Figure 11\get_result_A.m” to obtain results for panel A, B, and C of Figure 11. Use “Figure 11\get_result_B.m” to obtain results for panel D, E, and F of Figure 11. Run “Figure 11\Figure11.m” to replicate Figure 11.

12) To replicate Figure 12, run “Figure 12\welfare_cal.m” to compute market-level welfare gain and run “Figure 12\welfare.mxd” to draw Figure 12.

13) To replicate Figure 13, follow the C++ code in “code_main_model” to solve the model with subsidies. Then, use “Figure 13\calculation.m” to obtain results. Run “Figure 13\Figure13.m” to replicate Figure 13.

3. Replicate the results of the Online Appendix using code in "code_figure_table\OA"

1) To replicate Figure OA.1, enter folder "Figure OA1" and run "road_network.mxd".

2) To replicate Figure OA.2, enter folder "Figure OA2" and run "FigureOA2.m".

3) Data for Figure OA.3 are provided in "data\jiyan.mdb". Run "Figure OA3\ market_heterogeneity.mxd" to replicate Figure OA.3.

4) To replicate Figure OA.4, enter folder "Figure OA4" and run "FigureOA4.m".

5) To replicate Table OA2, enter folder "Table OA2"
   Run "CEM\check_balance.m" to replicate columns (1) and (2) of Table OA2.
   Run "NNM\check_balance.m" to replicate columns (3) and (4) of Table OA2.
   Run "PSM\check_balance.m" to replicate columns (5) and (6) of Table OA2.

6) To replicate Table OA3, enter folder "Table OA3"
   Run "employment\check_balance.m" to replicate columns (1) and (2) of Table OA3.
   Run "loan\check_balance.m" to replicate columns (3) and (4) of Table OA3.
   Run "income\check_balance.m" to replicate columns (5) and (6) of Table OA3.
   Run "entrepreneur\check_balance.m" to replicate columns (7) and (8) of Table OA3.

7) To replicate Table OA4, enter folder "Table OA4"
   Run "PSM\DID.do" to replicate column (1) of Table OA4.
   Run "CEM\DID.do" to replicate column (2) of Table OA4.
   Run "NNM\DID.do" to replicate column (3) of Table OA4.
   Run "SC_a_baseline_placebo\calculate_treatment_effect.m" to replicate the coefficients and "SC_a_baseline_placebo\placebo_test.m" to replicate the standard errors in column (4) of Table OA4.
   Run "SC_b_restricted_donor_pool\calculate_treatment_effect.m" to replicate the coefficients and "SC_b_restricted_donor_pool\placebo_test.m" to replicate the standard errors in column (5) of Table OA4.
   Run "SC_c_cross_validation\calculate_treatment_effect.m" to replicate the coefficients and "SC_c_cross_validation\placebo_test.m" to replicate the standard errors in column (6) of Table OA4.
   Run "SC_a_baseline_placebo\placebo_test.m" to replicate column (7) of Table OA4.

8) To replicate Table OA5, enter folder "Table OA5" and follow step 7).

9) To replicate Table OA6, enter folder "Table OA6" and follow step 7).

10) To replicate Table OA7, enter folder "Table OA7" and follow step 7).

11) To replicate Table OA8, enter folder "Table OA8" and run "reg.do".

12) To replicate Figure OA5, solve the occupation and borrowing decisions using the code in "code_main_model\6_implied_w_r". Set parameter values (including the interest rate, wages, and others) to those reported in the note of Figure OA5. Let the first market have d = 0 min and the second have d = 2 min.
Enter folder "Figure OA5" to run "wealth_prod.m" (produce results to draw panel A), "wealth_deposit_ratio.m" (produce results to draw panel B), and "draw.m" (draw the figure) in turn.

13) To replicate Figure OA6, solve the withdraw and deposit decisions using the code in "code_main_model\6_implied_w_r". Set parameter values (including the interest rate, wages, and others) to those reported in the note of Figure OA6. Let the first market have d = 0 min and the second have d = 2 min.
Enter folder "Figure OA6" to run "get_withdraw_deposit.m" (obtain policy function of withdraw and deposit), "wealth_prod.m" (produce results to draw panel A), "wealth_deposit_ratio.m" (produce results to draw panel B), and "draw.m" (draw the figure) in turn.

14) To replicate Figure OA7, enter folder "Figure OA7" and run "calculate.m" (obtain results) and "FigureOA7.m" (draw Figure OA.7) in turn.

15) To replicate Figure OA 8 and 9, solve the migration decisions using the code in "code_main_model\6_implied_w_r". Set parameter values (including the interest rate, wages, and others) to those reported in the note of Figure OA 8 and 9, respectively.
Enter folder "Figure OA8_9" to run "migration.m" (obtain policy function of migration), "result_OA8_A" (produce results to draw panel A of Figure OA8), "result_OA8_B" (produce results to draw panel B of Figure OA8), "result_OA9_A" (produce results to draw panel A of Figure OA9), "result_OA9_B" (produce results to draw panel B of Figure OA9), and "draw_OA8.m" and "draw_OA9" (draw the figure) in turn.

16) To replicate Figure OA10, enter folder "Figure OA10" and run "panel_A.m", "panel_B.m", "panel_C.m", and "FigureOA10" in turn.

17) To replicate Figure OA11, first update the model-predicted bank expansion choices "code_main_model\6_implied_w_r\input\comm_loc_model.mat". Then, enter folder "Figure OA11" and run "1_GIS_bank_location_data.mxd" for panel A and "2_GIS_bank_location_predict.mxd" for panel B.

18) To replicate Figure OA12, enter folder "Figure OA12" and run "draw_AD.m", "draw_BE.m", "draw_CF.m", and "FigureOA12" in turn.

19) To replicate Figure OA13, enter folder "Figure OA13" and run "zoom_in_bangkok_baseline.mxd" for panel A and "zoom_in_bangkok_infcapacity.mxd" for panel B.

20) To replicate Figure OA14 and 15, enter folder “Figure OA14_15” and run “data87.mxd” for panel A of Figure OA14. Panels B to J of Figure OA14 and Figure OA15 can be drawn by modifying layer “new branch, 87 (data)” of “data87.mxd”.

21) To replicate Figure OA16, enter folder “Figure OA16” and run “calculate.m” and “FigureOA16.m” in turn.

22) To replicate Table OA11, enter folder “Table OA11” and run “count_num.m”.
Follow the steps in section 1 of this readme file to solve the model without productivity heterogeneity. The only change needed is to use variable “Z_uniform” instead of “Z” (in “data\input.mat”) as input.

23) To replicate Figure OA17, enter folder “Figure OA17” and run “1986.mxd” for panel A of Figure OA17. Panels B to K of Figure OA17 can be drawn by modifying layer “1986” of “1986.mxd”.

24) To replicate Figure OA18, enter folder “Figure OA18” and run “1986.mxd” for panel A of Figure OA18. Panels B to K of Figure OA18 can be drawn by modifying layer “1986” of “1986.mxd”.

25) To replicate Figure OA19, 21, enter folder “Figure OA19_21” and run “FigureOA19.m” and “FigureOA21.m” in turn.

26) To replicate Figure OA20, enter folder “Figure OA20”. First calculate bank profit using cal_profits.m. Then, run “figure.mxd”.

27) To replicate Table OA12, enter folder “Table OA12”.
    i. Run “a_construct_data_for_simulation.m” to prepare data for simulation. The output is saved in folder “simulation_input”.
    ii. Copy the files (bank_opt.dat, comm_loc_model.dat, distance.dat, pdf_ini.dat, Pi.dat, r.dat, w.dat, Z.dat) into folder “code_main_model\7_simulate_perfect_DID\1_perfect_match\input”, “code_main_model\7_simulate_perfect_DID\2_anticipation\input”, and “code_main_model\7_simulate_perfect_DID\3_lag\input” as future input.
    iii. Run C++ code in “code_main_model\7_simulate_perfect_DID\1_perfect_match”, “code_main_model\7_simulate_perfect_DID\2_anticipation”, and “code_main_model\7_simulate_perfect_DID\3_lag” and the results are saved in folder “computation_results”. Run “computation_results\read_data.m” in each folder to read and save the results.
    iv. Go back to folder “\code_figure_table\OA\Table OA12” and run “b_preparedata.m” to construct data for DID. Use line 25 – 27 to read results from different folder. The file panel_all.xslx will be produced.
    v. Run “DID.do” to run DID regression.

28) To replicate Figure OA22, first use C++ code in "code_main_model\6_implied_w_r" to solve the model with parameter values specified in the note of Figure OA22. In particular, we only solve for two markets. Then, use "Figure OA22\FigureOA22.m" to read results and draw Figure OA22.

29) To replicate Figure OA23, first vary κ from 1.6 to 3.4, use the code in "code_main_model" to solve the model, and use the code in "code_figure_table\maintext\Table 2\model\basic_stats.m" to compute the migration moments. Second, vary η from 0 to 0.024, use the code in "code_main_model" to solve the model, and use the code in "code_figure_table\maintext\Table 2\model\basic_stats.m" to compute the migration moments. Finally, use "Figure OA23\FigureOA23.m" to read the results and draw Figure OA23.

30) To replicate Figure OA24, first recalibrate the model to have migration flow of 0% and 14.1%, respectively following section 1 of this readme file. Then, use "maintext\Figure 8\calculation_aggregate.m" to calculate the aggregate dynamics.

31) To replicate Figure OA25, 26, use "maintext\Figure 9_10\model\calculate_gini.m" to calculate the results.

32) To replicate Figure OA27, use "maintext\Figure 12\welfare_cal.m" to calculate the results.

33) To replicate Figure OA28, first recalibrate the model with $\theta_\kappa = 0.03$ and $\theta_\kappa = 0.05$, respectively following section 1 of this readme file. Then, use "Figure 8\calculation_aggregate.m" to calculate the aggregate dynamics.

34) To replicate Figure OA29, follow the C++ code in "code_main_model\6_implied_w_r" to solve the model with only credit channel, with only deposit channel, and no reform. Enter folder "Figure OA29" and use "avg_z.m" to calculate average talent weighted by output, use "additional_fund_flow.m" to calculate additional flow of funds, and use "FigureOA29.m" to draw Figure OA29.

35) To replicate Figure OA30, first recalibrate the model with $\theta_\zeta = 0$ following section 1 of this readme file. Second, solve the model with only credit channel and with only deposit channel, holding the locations of new branches unchanged. Finally, follow the steps to get the DID coefficients of Table 2 in the main text.

36) To replicate Figure OA31, use "Figure 8\calculation_aggregate.m" to calculate the aggregate dynamics.

37) To replicate Figure OA32, first recalibrate the model with $\theta_\zeta = \theta_\psi$ following section 1 of this readme file. Second, solve the model with only credit channel and with only deposit channel, holding the locations of new branches unchanged. Finally, follow the steps to get

the DID coefficients of Table 2 in the main text.

38) To replicate Figure OA33, use "Figure 8\calculation_aggregate.m" to calculate the aggregate dynamics.

39) To replicate Figure OA34, first recalibrate the model with $Z_i = 1$ for all markets following section 1 of this readme file. (The only change needed is to use variable "Z_uniform" instead of "Z" (in "data\input.mat") as input.) Then, follow the steps to get the DID coefficients of Table 2 in the main text.

40) To replicate Figure OA35, use "Figure 8\calculation_aggregate.m" to calculate the aggregate dynamics.

41) To replicate Figure OA36, 37, use "Figure 9_10\model\calculate_gini.m" to calculate the results.

42) To replicate Table OA13, enter folder "Table OA13" and run "fill_table.m".
Follow the steps in section 1 of this readme file to solve the model with intramarket distances. The market area and intramarket distance data are saved in "data\area.mat". When solving the model, read intramarket distance data ("\data\area.mat") into the matlab code and add it to the distance from nearest branch as shown by equation (4.1) of the Online Appendix. An example is given by lines 4-5 of file "code_main_model\1_solve_ss\input\prepare_input.m"

43) To replicate Figure OA38, first recalibrate the model with $\xi = 0.05$ following section 1 of this readme file. Second, recalibrate the model with $\theta_\psi = 0.36$ following section 1 of this readme file. Finally, follow the steps to get the DID coefficients of Table 2 in the main text.

44) To replicate Figure OA39, use "Figure 8\calculation_aggregate.m" to calculate the aggregate dynamics.

45) To replicate Figure OA40, first recalibrate the model with $\gamma = 0.7$ following section 1 of this readme file. Second, recalibrate the model with $\rho = 4.0$ following section 1 of this readme file. Finally, follow the steps to get the DID coefficients of Table 2 in the main text.

46) To replicate Figure OA41, use "Figure 8\calculation_aggregate.m" to calculate the aggregate dynamics.

47) Table OA.14 is reported in "Table OA14".

48) To replicate Figure OA42, first recalibrate the model with capital flows reported in Table OA.14 following section 1 of this readme file. Then, use "Figure 8\calculation_aggregate.m" to calculate the aggregate dynamics.

49) To replicate Figure OA43, 44, first solve the "branch as in data" case using C++ code in "code_main_model\6_implied_w_r" taking branch location in the data as given. Second, solve the "random branch" case using C++ code in "code_main_model\6_implied_w_r" using random branch locations. Finally, use "Figure 9_10\model\calculate_gini.m" to calculate the results.

50) To replicate Figure OA45, use "maintext\Figure 12\welfare_cal.m" to calculate the results.

51) To replicate Figure OA46, use "Figure 8\calculation_aggregate.m" to calculate the aggregate dynamics.

52) To replicate Figure OA47, enter folder "Figure OA47" and run "random_assignment.m", "timing_difference_random_assignment.m", and "FigureOA47.m" in turn.

53) To replicate Figure OA48, run "Figure OA48\flow_of_fund.m" to obtain results. Then, run "flow_of_fund.mxd" to draw the figure.

54) To replicate Figure OA.49, run "Figure OA49\FigureOA49.m".

55) To replicate Figure OA50, run C++ code in "code_main_model\6_implied_w_r". Specifically, use the code to only solve the transition of one market by setting variable nmkt to 1 and set the other parameter values consistent with the table note of Figure OA.50. Then, run "Figure OA50\close_branch.m" to draw the figure.

56) To replicate Figure OA51, first solve the main model following section 1 of this readme file, allowing banks to open new branches till 2011. Then, use "Figure OA51\calculation_aggregate.m" to obtain results.

57) To replicate Figure OA52, run C++ code in "code_main_model\6_implied_w_r". Specifically, use the code to only solve the transition of one market by setting variable nmkt to 1 and set the other parameter values consistent with the table note of Figure OA.52. Then, run "Figure OA52\open_branch.m" to draw the figure.

58) To replicate Figure OA53, first solve the main model following section 1 of this readme file, introducing expected shock in ξ. Then, use "Figure OA51\calculation_aggregate.m" to obtain results.

59) To replicate Figure OA54, first solve the main model following section 1 of this readme file, introducing expected shock in ψ. Then, use "Figure OA51\calculation_aggregate.m" to obtain results.

60) To replicate Figure OA55, first solve the main model following section 1 of this readme file, introducing expected shock in $\varepsilon_{z,t}$. Then, use "Figure OA51\calculation_aggregate.m"

to obtain results.

61) To replicate Figure OA56, use the data from “data\jiyan.mdb” and run “Figure OA56\partition.mxd”.

62) To replicate Figure OA.57, run “Figure OA.57\Figure OA.57.m”.

63) To replicate Figure OA.58, run “Figure OA.58\Figure OA.58.m”.

64) Table OA.15 is calculated and reported in “Table OA15”.