**Instructions for Running Computational Model and Simulation**

The following provides information for running the computational model and simulation reported in the accompanying paper. All code is provided to the user “as is” and may be freely manipulated or distributed. **Please cite the original paper if any portions of the model and/or simulation are borrowed, adapted, or used for other purposes**. There is no expectation for the author to provide technical guidance or assistance with troubleshooting, running, or adapting any portions of the provided code.

1. Extract all files in the STCompModel.zip folder into the ***same directory/folder***. There should be 2 files:
	1. OrgImpactST\_model.R
	2. OrgImpactST\_simulation.R
2. If necessary, download and install R (<https://cran.r-project.org/>)
3. If they have not been installed previously, the following packages must be installed by running the following commands in the R console window:

install.packages(“parallel”)

install.packages(“snowfall”)

install.packages(“rlecuyer”)

1. The OrgImpactST\_model.R file provides the computational model code summarized in Table 7 and whose steps are detailed in Appendix B of the paper. The model code is written as a function named STinTraining. To run the model under a single combination of parameter values, perform the following steps:
	1. Open OrgImpactST\_model.R.
	2. Select and run all lines. This will create a new object in the global R environment named STinTraining.
	3. STinTraining is a function defined by six input parameters. Default values are included for each parameter:
		1. time: the number of time steps the model will run. Default value = 30
		2. org: the total number of organizations to simulate (# of orgs with ST AND # of orgs without ST). Default value = 500
		3. emp: the total number of employees per organization. Default value = 100
		4. volTurn: the voluntary turnover rate (must be value between 0 and 1). Default value = .10
		5. STeffectSize: the relative difference in performance rates due to learning between employees experiencing vs. not experiencing ST (must be value between 0 and +∞). In the paper, this value is defined as θ. When θ = 1, there will be no differences in the performance improvement rates between ST and non-ST employees. When θ > 1, the performance improvement rates of ST employees will be slower than those of non-ST employees. Default value = 1.5
		6. muRateNST: the rate of performance improvement due to learning for employees not experience ST (value should be < 1 to conform to decreasing power law). Default value = -1.5
	4. To run the model using the default values, run the following in the R console window:

sim <- STinTraining()

* 1. Running the line of code above will create a new object in the global R environment named sim that contains the results of the model run. The sim object is a list containing four data frames (listed in order):
		1. sim[[1]]: Aggregate summary statistics (mean, SD, and effect sizes) of conditions for each time period
		2. sim[[2]]: Data for all simulated employees remaining at final time period
		3. sim[[3]]: Performance potential for all organizations at each time period
		4. sim[[4]]: Miscellaneous summary statistics calculated for each time period (primarily useful for testing that model specifications worked as intended)
	2. To run the model using different values from the default, simply provide alternative parameter values to the STinTraining function. For example, the following will run the model with 15 time steps, 50 total organizations, 10 employees per organization, a 15% voluntary turnover rate, a 2x relative difference in performance improvement rates, and performance improvement rate of -2 for non-ST employees:

sim <- STinTraining(time = 15, org = 50, emp = 10, volTurn = .15, STeffectSize = 2, muRateNST = -2)

1. The OrgImpactST\_simulation.R file contains code for creating an experimental simulation using the computational model. To run simulations using this file:
	1. Open OrgImpactST\_simulation.R
	2. Create the simulation conditions to be run:
		1. Line 6 allows the user to specify the parameters and parameter values to manipulate in each simulated condition. The current default setting will run the 20 simulated conditions reported in the paper.
		2. The experimental conditions are recorded in a matrix named conds. Each row of the matrix corresponds to an experimental condition.
		3. **Important**: Only the six input parameters described in step 4.c are able to be manipulated as unique simulation conditions. Users wishing to include additional factors or additional levels of a factor in a simulation will need to update lines 6 and lines 25 of the code.
	3. Select and run lines 10-30 to run the simulation.
	4. When completed, the simulation will create a new object in the global R environment named condsDat. condsDat is a list object. Each element of the list contains all the output from a single experimental condition (i.e., condsDat[[1]] contains all the data for the first experimental condition; condsDat[[15]] contains all the data for the 15th experimental condition, etc.). The numbering of the experimental conditions corresponds to the row numbers of conds as described in step 5.b.ii.
2. The remaining code provided on lines 37-100 of this file construct and compile new data files from condsDat and provide exemplars for creating various plots for exploring the data. Note that these lines of code are written to be run with the 20 simulation conditions reported in the paper. Users wishing to alter the number of factors or number of levels per factor included in the simulation will need to adjust this code accordingly.