Documentation on Running Experiments with WOLVES Model

WOLVES model is implemented using the COSIVINA framework, a free object-oriented framework to construct and simulate dynamic field models in MATLAB 2016b and later versions (Note: in 2021, we also introduced a python-based version). WOLVES provides a graphical user interface to view the model's looking behaviours in real-time, the activation time courses of different fields and to allow adjusting model parameters online.

To get started with WOLVES in MATLAB, follow the steps below. You might also want to check out our visual guide online https://dynamicfieldtheory.org/dft_case_studies/wolves/.

Step 1. Download the project data from our OSF site: https://osf.io/ng2vf/. To do this, click on the 'OSF Storage (United States)' text in the 'Files' section. Then click 'Download as zip'. Once downloaded, you can unzip the folder and save someplace convenient on your computer.

Alternatively, you can clone our published repository of completed DFT projects from GitHub. If you do this via Github, the project folder is ***Generalizing_WOLVES_Bhat_2023***, but you'll also get the original WOLVES repository as a bonus in ***WOLVES_PsychReview_2021***. Both folders contain the WOLVES model, experiment, and analysis code scripts.

Note that the Github version does not contain all the simulation data as these files are too big to store on Github. Thus, if you want the simulation data, use the OSF route.

Step 2. Make sure you have ***COSIVINA*** and ***jsonlab*** downloaded. ***COSIVINA*** is a library supporting the underlying DFT framework. You can clone this from our Github repo: https://github.com/cosivina/cosivina. You'll also need to download ***jsonlab*** support; this is required to support loading and saving parameter files in json format. You can clone this from the following Github repo: https://github.com/fangq/jsonlab.

Step 3. Once you've downloaded ***COSIVINA*** and ***jsonlab***, you should have a 'cosivina' folder in your Github folder. You should also have a 'jsonlab' folder in your Github folder. Navigate to the 'cosivina' folder in MATLAB and run '*setpath.m'*. This will add the COSIVINA library to your current MATLAB session. If successful, you should see something like the following message displayed in your MATLAB window:

"All required directories added to path.
Directory '.../GitHub/jsonlab' added for JSONlab support."

Step 4: Inside MATLAB's folder view, navigate to the osf folder you downloaded and right click on this folder and select 'add to path' and then choose 'selected folder and subfolders'. Subfolder name ***wolves_core*** contains the main scripts implementing the model. Subfolders ***experiments_code*** and ***analysis_code*** contain the scripts implementing the different experiments and corresponding analyses respectively. Subfolder ***support_code*** contains some supporting functions called in scripts within ***experiments_code*** and ***analysis_code*** folders.

Note: if you downloaded the model from Github, add the '***Generalizing_WOLVES_Bhat_2023'*** folder and subfolders to your path.

Step 5. Within MATLAB, navigate into the ***wolves_core*** subfolder. This contains the following files:

1. *XSIT_Manual_run.m* : This is the main file used to instantiate and run the model.
2. *createComboSim.m* : This file creates the model and details its architecture (fields, interactions, and inputs).
3. *createComboGUI.m* : This file creates and initializes the GUI for the model
4. *createComboControls.m* : This creates and initializes GUI controls
5. *wolvesPaperPR.json* : This json file contains the parameter values for the model to use.

Step 6. Open the *XSIT_Manual_run.m*, push the *Run* button in MATLAB and you should see the model GUI open up. This file is set up to run Experiment 1 from Mather, Schafer & Houston-Price (2011) – the first experiment we simulate in the paper. Note that output from the model run will be dumped to whatever folder you happen to be within in the MATLAB window. As good practice, before hitting *Run*, we usually navigate to the main folder (either the 'osf' folder you downloaded or '*Generalizing_WOLVES_Bhat_2023'*) so the model results are placed in a readily visible place.

Further Details

Inside *XSIT_Manual_run.m*:

- The model can be run in 3 modes: a GUI mode, non-gui *single-run* mode, non-gui *batch mode* that uses parallel computing (and can be run on a high performance computer cluster or HPC). Set the `mode` variable to any of the following valid values (around line *5*).

Mode = 1 [*Default setting]*: Run a GUI to visualize the model behaviour in real time. The GUI shows real-time activity in different fields of the model as well as the looking behaviour of the model over time in the looking time plot. The model displays/records looking at 5 different spatial positions, these looking positions are specified as: Left (red solid curve in the looking time plot), Left-middle (red dotted curve), Middle (green curve), Right-middle (blued dotted curve) and Right (blue solid curve). The GUI mode gives you the ability to change and play with the parameters for behaviour evaluation and parameter optimization (see the buttons in the bottom right).

Mode = 2: Using an HPC or multi-core computer, run a batch a simulations/subjects in an experiment in the model without any GUI. This requires changing the 'numSubjects' ***for*** loop in the corresponding experiment file to ***parfor*** for parallel computation. Note that the model takes about 8 hours to run a batch of 300 simulations on an HPC with 96 cores. It takes over 24 hours to run a batch of simulations on a mac pro laptop with 6 cores.

  - If you want to run a fresh batch of simulations, change mode = 2 in XSIT_Manual_run.m (line 5). Then open, for instance, 'MSHP_Exp1_2011.m' in the 'experiments_code' folder and change line 22 so it says 'parfor subject = 1:numSubjects' instead of 'for subject = 1:numSubjects'

Mode = 0: runs one simulation at a time without GUI on a standalone CPU. This is not recommended as it takes forever to run a batch of 300 simulations.

- **`gui_speed`** variable determines how fast the GUI should update. Ideal values range between 1 to 20. The visualizations update only every *gui-speed* timesteps.
- To modify the memory *tau_Build* and *tau_Decay* parameters, change parBuild and parDecay on line 29. These parameters are changed with ages in the main paper (see details below).
- Each empirical task has associated one or more corresponding scripts for simulation and a script for analyses of the simulation(s). Tasks and their corresponding simulation and analysis files are listed below.
- All these experiments are listed from around line *39* in *XSIT_Manual_run.m*. To choose an experiment to run, use the experiment corresponding `taskvar` variable value (around line 37 in file) as indicated in the table below. Open the corresponding experiment script file and ensure the first loop (i.e., the subject loop) is a ***parfor*** loop if you set **`mode=2`** for parallel simulations. Else for a single run with GUI on, i.e mode 1, it should be an ordinary ***for*** loop.
- Switch to *XSIT_Manual_run.m*. Press the run button in MATLAB and your simulation should be running!

- Some tasks require multiple simulations to be run as mentioned in the table:

Exp. No.	Task/Experiment	Experiment/Simulation Script file (Inside ***experiments_code*** folder)	Analysis Script file (Inside ***analysis_code*** folder)
14	Mather Schafer Houston-Price (2011) Experiment 1	`MSHP_Exp1_2011.m` (use `taskvar = 14`) Enable noise parameters in *XSIT_Manual_run* file at line 33. Two separate sims are needed of this task corresponding to each age group, one for silent condition, (use `Labelling_condition_ON= 0`, at line 42) and another for labelling condition (use `Labelling_condition_ON= 1`) This experiment uses two age groups, hence varying memory parameters on line 29: 9-14 month [Younger]: (tau_Build = 1200, tau_Decay = 800) 15-19 month [Older]: (tau_Build = 1200, tau_Decay = 4500)	dataAnalysis_MSHP_2011.m

15	Mather & Plunkett (2012) Experiment 1	Mather_Plunkett_Exp1_2012.m (use `taskvar = 15`) Enable noise parameters in *XSIT_Manual_run* file at line 33 and memory parameters are (tau_Build = 1200, tau_Decay = 8000)	dataAnalysis_Mather_Plunkett_Exp1_2012
16	Mather Schafer Houston-Price (2011) Experiment 2	MSHP_Exp1_2011.m (use `taskvar = 16`) Enable noise parameters in *XSIT_Manual_run* file at line 33. Two separate sims are needed of this task corresponding to each age group, one for silent condition, (use `Labelling_condition_ON= 0`, at line 56) and another for labelling condition (use `Labelling_condition_ON= 1`) This experiment uses two age groups, hence varying memory parameters: 16-month [Younger]: (tau_Build = 1200, tau_Decay = 4500) 21-month [Older]: (tau_Build = 1200, tau_Decay = 8000)	dataAnalysis_MSHP_2011.m

- Once the simulation(s) necessary for a task are complete, analysis can be conducted. Open the ***analysis_code*** folder. Look for the corresponding data analysis file as listed in the table above. Open the data analysis file, navigate back to the folder where your simulation mat file(s) is/are located.
- Inside the dataAnalysis file: Update the `simName` variable with the name(s) of your simulation(s) and press run. The script will produce an output excel file with RMSE and MAPE values for the simulation(s), display these values in command window as well as show plots associated with the task.
- Note that if you want to generate the random permutations as used to compute the standard errors for the model in the paper, use the analysis file that end with ‘randpermforfigures’.