

Script Notes: Figure 2 Generation

This document explains how to run the scripts used to generate Figure 2 in the manuscript.

Prerequisites

Before running any script in this directory, you **must** call `setup_paths.m` to add the `src/` directory to the MATLAB path:

```
setup_paths();
```

This function: - Locates the repository root relative to the `scripts/` directory - Adds `src/` (and all subdirectories) to the MATLAB path - Errors if the `src/` directory is not found

Note: Each Figure 2 script calls `setup_paths()` automatically at the start, so you only need to call it manually if running individual functions interactively.

Figure 2 Scripts

1. Single vs Dual Adaptation Example

File: `Fig_2_single_vs_dual_adaptation_example.m`

Purpose: Compares SFA-only vs SFA+STD on identical networks

This script runs two simulations with the **same connectivity matrix \mathbf{W}** and **same external stimulus** (controlled by `rng_seeds = [42 42]`), allowing direct comparison of adaptation mechanisms:

Run	n_a_E	n_b_E	Description
1	3	0	SFA only (3 timescales)
2	3	1	SFA + STD

Outputs: - Combined 6-panel time series figure showing: external input, dendritic state, synaptic output, adaptation, STD, and local Lyapunov exponent - Individual run figures saved to condition-specific subfolders

Output Location:

```
figs/srnn_comparison_<YYYYMMDD_HHMM>/  
+-- SFA_only/          # Individual SFA-only plots  
+-- STD_and_SFA/       # Individual SFA+STD plots  
+-- combined_comparison.{fig,svg,png,jp2}
```

2. Fraction Excitatory Analysis

File: `Fig_2_fraction_excitatory_analysis.m`

Purpose: Parameter space sweep over excitatory fraction (f)

This script uses `ParamSpaceAnalysis` to systematically vary: - f (fraction excitatory): 0.4 to 0.6 in 5 levels - **reps** (repetitions): 5 independent networks per parameter combination

Each combination is tested under all **four adaptation conditions**: - `no_adaptation` - `sfa_only` - `std_only` - `sfa_and_std`

Outputs: - Histograms of mean synaptic output and LLE across conditions - Paired swarm plots comparing LLE during stim vs no-stim periods - Combined 3×4 panel figure

Data Output Location:

```

data/param_space/param_space_<note>_nLevs_<N>_<timestamp>/
+-- param_space_summary.mat
+-- psa_object.mat
+-- no_adaptation/
|   +-- param_space_results_no_adaptation.mat
+-- sfa_only/
|   +-- param_space_results_sfa_only.mat
+-- std_only/
|   +-- param_space_results_std_only.mat
+-- sfa_and_std/
    +-- param_space_results_sfa_and_std.mat

```

Figure Output Location (when `save_figs = true`):

```

figs/fraction_excitatory_analysis/
+-- fraction_excitatory.{fig,svg,png,jp2}
+-- data_source.txt      # Records which data folder figures came from

```

Configuration Options

Both scripts have configuration flags at the top:

Variable	Default	Description
<code>save_figs</code>	<code>true / false</code>	Whether to save figures to disk
<code>save_workspace</code>	<code>false</code>	Whether to save full workspace (very large files – only needed for e.g. debug)

Handling Interrupted Runs

If `Fig_2_fraction_excitatory_analysis.m` is interrupted before completion, the `ParamSpaceAnalysis` class can consolidate partial data for analysis and plotting:

```

psa = ParamSpaceAnalysis();
psa.output_dir = '/path/to/interrupted/run';
psa consolidate(); % Merge completed batch files

```

Note: This consolidates data from completed batches but does **not** resume computation. To obtain complete results, you must restart the analysis from the beginning.

Batch checkpoint files are stored in `temp_batches/` within the output directory and are cleaned up after successful consolidation.

See Also

- `Stability_Analysis_Code_Structure.md` – Class documentation
- `parameter_table.md` – Parameter reference