

Instruction for Validation

These data and code are provided by Baishen Liang in the Institute of Psychology, Chinese Academy of Sciences (liangbs@psych.ac.cn or liangbs95@gmail.com). We welcome validation of data and code, and are happy if they are also useful in your research work. Please inform us and get approval before using any materials provided here (except Matlab functions provided by the product) to publish your work or in any commercial actions.

Functional localization pretest (.\\fMRI pretest)

Results (BOLD activation maps) for the functional localization pretest are provided in the fMRI pretest document. The **A-D_Clustsim.thr.preCG.nii** is the activation map for the “say AH” condition (phonation/ laryngeal motor cortex); the **D-A_Clustsim.thr.preCG.nii** is the activation map for the “say D” condition. The **tpl-MNI152NLin6Asym_res-02_desc-brain_T1w.nii.gz** is the template structural image. The **TBS Target Ibsonly.txt** includes the MNI coordinates for our selected LMC and TMC targets.

You can use fMRI visualization software (e.g., MRIcron or Mango) to scan through the image.

Experiment 1: psychometric curve fitting (.\\Exp1_slope)

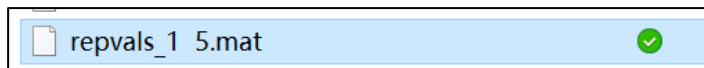
Code for Experiment 1 need to run for 6 times (3 groups of trials by ambiguity in the unattended dimension × twice for each). Here I demonstrate the procedure for running the “unambiguous” group of trials (Figure 1A).

Step 1. Set the path to “[codes_for_dataprocess\\exp1\\code](#)”;

Step 2. Run this function to prepare values for replacing slopes in TMS conditions that are deleted (for details, see the *Methods* section in the paper).

```
bsliang_TMS_main('unambiguous',1);
```

Step 3. After finishing step2, a new *.mat file appears in the “data” path:

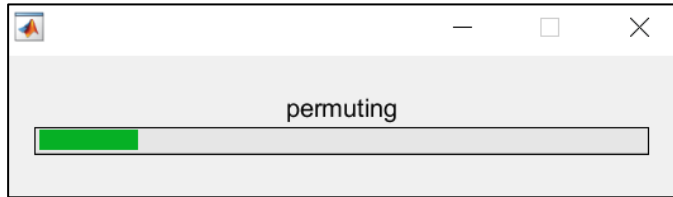


If you want to redo the step2, please delete this file.

Step 4. Run this function to do curve fitting, slope extraction, plotting, and statistics:

```
bsliang_TMS_main('unambiguous',0);
```

This box prompts out and permutation starts. Wait until it is filled.



Step 5. As the program finishes, results are directly displayed in the interactive window
(red box for significant results):

```

=====Left Clear=====
Tone Larynx:      M = 0.333, rank = 91324 p = 0.913 Cohen's d = 0.234
Tone Tongue:      M = 0.386, rank = 95362 p = 0.954 Cohen's d = 0.301
Tone Larynx - Tongue: MD = -0.108, Z = -0.73258, Ranksum = 239, p = 0.23191 Cohen's d = -0.
Phon Larynx:      M = -0.632, rank = 524 p = 0.00525 Cohen's d = -0.459
Phon Tongue:      M = -0.442, rank = 552 p = 0.00553 Cohen's d = -0.467
Phon Larynx - Tongue: MD = -0.163, Z = -0.89339, Ranksum = 230, p = 0.18582 Cohen's d = -0.
=====Right Clear=====
Tone Larynx:      M = 0.117, rank = 70166 p = 0.702 Cohen's d = 0.101
Tone Tongue:      M = 0.142, rank = 70296 p = 0.703 Cohen's d = 0.107
Tone Larynx - Tongue: MD = -0.077, Z = -1.0921, Ranksum = 132, p = 0.13739 Cohen's d = -0.0
Phon Larynx:      M = -0.211, rank = 20236 p = 0.202 Cohen's d = -0.161
Phon Tongue:      M = -0.013, rank = 47724 p = 0.477 Cohen's d = -0.013
Phon Larynx - Tongue: MD = -0.253, Z = -1.2953, Ranksum = 124, p = 0.097609 Cohen's d = -0.
=====Left Noise=====
Tone Larynx:      M = 0.062, rank = 60046 p = 0.6 Cohen's d = 0.043
Tone Tongue:      M = 0.554, rank = 98413 p = 0.984 Cohen's d = 0.394
Tone Larynx - Tongue: MD = -0.487, Z = -1.6796, Ranksum = 186, p = 0.04652 Cohen's d = -0.3
Phon Larynx:      M = 0.171, rank = 69261 p = 0.693 Cohen's d = 0.096
Phon Tongue:      M = -0.074, rank = 38291 p = 0.383 Cohen's d = -0.057
Phon Larynx - Tongue: MD = 0.352, Z = 1.1175, Ranksum = 219, p = 0.86811 Cohen's d = 0.162
=====Right Noise=====
Tone Larynx:      M = -0.473, rank = 4717 p = 0.0472 Cohen's d = -0.333
Tone Tongue:      M = -0.239, rank = 23252 p = 0.233 Cohen's d = -0.149
Tone Larynx - Tongue: MD = -0.221, Z = -0.3767, Ranksum = 148, p = 0.3532 Cohen's d = -0.13
Phon Larynx:      M = -0.302, rank = 25280 p = 0.253 Cohen's d = -0.16
Phon Tongue:      M = -0.081, rank = 42489 p = 0.425 Cohen's d = -0.049
Phon Larynx - Tongue: MD = -0.208, Z = -0.2844, Ranksum = 62, p = 0.38805 Cohen's d = -0.08

```

```
q_fdr_output =
```

```

0.9841
0.9841
0.0332
0.0332
0.7030
0.7030
0.5056
0.5727
0.9841
0.9841
0.9841
0.7658
0.5056
0.5056
0.5056
0.5665
0.5566
0.5566
0.5056
0.5056
0.1861
0.9841
0.5665
0.5665

```

(Tone refers to tone; Phon refers to consonant)

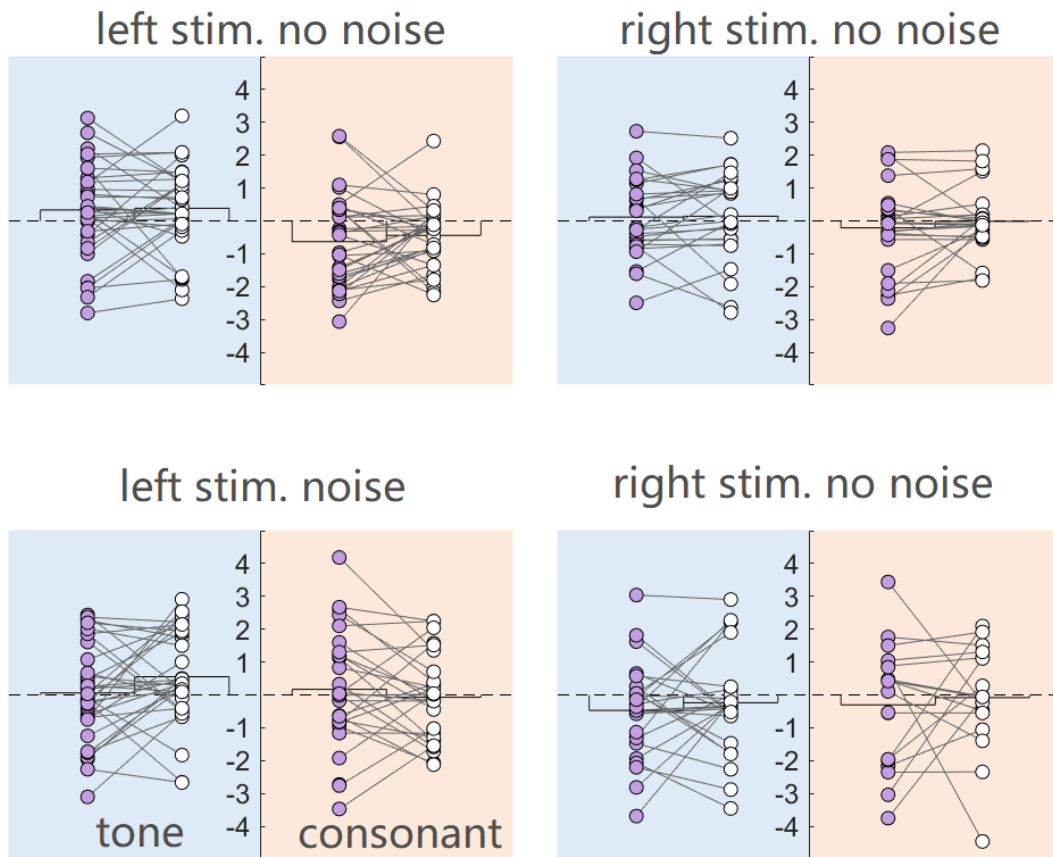
Upper are statistic results before FDR correction, lower are p-values corrected by FDR correction.

Note that the output for q_fdr_output is in a different order than statistic results. The corresponding conditions are shown below (red color with asterisk for significant results):

q_fdr_output	Condition
0.9841	Left Hemisphere, no noise, larynx stim., tone
0.9841	Left Hemisphere, no noise, tongue stim., tone
0.0332*	Left Hemisphere, no noise, larynx stim., consonant
0.0332*	Left Hemisphere, no noise, tongue stim., consonant
0.7030	Right Hemisphere, no noise, larynx stim., tone
0.7030	Right Hemisphere, no noise, tongue stim., tone
0.5056	Right Hemisphere, no noise, larynx stim., consonant
0.5727	Right Hemisphere, no noise, tongue stim., consonant
0.9841	Left Hemisphere, noise, larynx stim., tone
0.9841	Left Hemisphere, noise, tongue stim., tone
0.9841	Left Hemisphere, noise, larynx stim., consonant
0.7658	Left Hemisphere, noise, tongue stim., consonant
0.5056	Right Hemisphere, noise, larynx stim., tone
0.5056	Right Hemisphere, noise, tongue stim., tone
0.5056	Right Hemisphere, noise, larynx stim., consonant
0.5665	Right Hemisphere, noise, tongue stim., consonant
0.5566	Left Hemisphere, no noise, larynx-tongue stim., tone
0.5566	Left Hemisphere, no noise, larynx-tongue stim., consonant
0.5056	Right Hemisphere, no noise, larynx-tongue stim., tone
0.5056	Right Hemisphere, no noise, larynx-tongue stim., consonant
0.1861	Left Hemisphere, noise, larynx-tongue stim., tone
0.9841	Left Hemisphere, noise, larynx-tongue stim., consonant
0.5665	Right Hemisphere, noise, larynx-tongue stim., tone
0.5665	Right Hemisphere, noise, larynx-tongue stim., consonant

Step 6. Plots are shown directly:

The raw plots were displayed without texts. Below shows the order of plots.



(purple: LMC stimulation; white: TMC stimulation)

Step 7. Repeat the steps above for half-ambiguous and ambiguous trial groups.

```
bsliang_TMS_main('halfambiguous',1)
```

```
bsliang_TMS_main('halfambiguous',0)
```

```
bsliang_TMS_main('ambiguous',1)
```

```
bsliang_TMS_main('ambiguous',0)
```

Experiment 2: psychometric curve fitting (.\Exp2_slope)

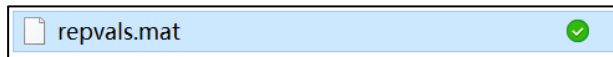
Code for Experiment 2 need to run for 2 times (once for making values to replace invalid data, and once for processing TMS data). Here I demonstrate the procedure for running (Figure 3).

Step 1. Set the path to “`codes_for_dataprocess\exp2\code`”;

Step 2. Run this function to prepare values for replacing slopes in TMS conditions that are deleted (for details, see the *Methods* section in the paper).

```
bsliang_TMS_main(1);
```

Step 3. After finishing step2, a new *.mat file appears in the “`data`” path:

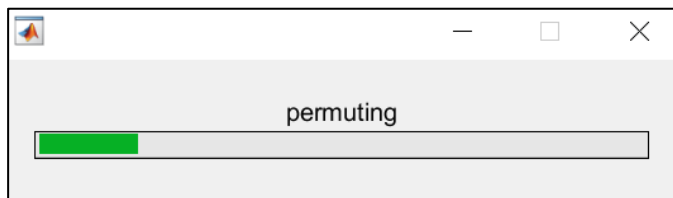


If you want to redo the step2, please delete this file.

Step 4. Run this function to do curve fitting, slope extraction, plotting, and statistics:

```
bsliang_TMS_main(0);
```

This box prompts out and permutation starts. Wait until it is filled.



Step 5. As the program finishes, results are directly displayed in the interactive window (red boxes for significant results, green boxes and text for conditions):

Tone iTBS:	M = 0.17, rank = 17586 p = 0.176 Cohen's d = 0.188	Left Clear
Tone cTBS:	M = 0.418, rank = 97814 p = 0.978 Cohen's d = 0.43	
Tone iTBS - cTBS:	MD = -0.25, Z = -1.6682, Signrank = 101, p = 0.95236 Cohen's d = -0.3	
Phon iTBS:	M = -0.164, rank = 79502 p = 0.795 Cohen's d = -0.167	
Phon cTBS:	M = -0.368, rank = 1217 p = 0.0122 Cohen's d = -0.474	Right Clear
Phon iTBS - cTBS:	MD = 0.2, Z = 1.4799, Signrank = 218, p = 0.069453 Cohen's d = 0.22	
Tone iTBS:	M = 0.053, rank = 35193 p = 0.352 Cohen's d = 0.078	
Tone cTBS:	M = 0.473, rank = 98910 p = 0.989 Cohen's d = 0.489	
Tone iTBS - cTBS:	MD = -0.42, Z = -2.1526, Signrank = 83, p = 0.98432 Cohen's d = -0.49	Left Noisy
Phon iTBS:	M = -0.11, rank = 71440 p = 0.714 Cohen's d = -0.113	
Phon cTBS:	M = -0.152, rank = 22961 p = 0.23 Cohen's d = -0.152	
Phon iTBS - cTBS:	MD = 0.04, Z = 0.21526, Signrank = 171, p = 0.41478 Cohen's d = 0.05	
Tone iTBS:	M = 0.116, rank = 37126 p = 0.371 Cohen's d = 0.077	Right Noisy
Tone cTBS:	M = -0.761, rank = 575 p = 0.00576 Cohen's d = -0.687	
Tone iTBS - cTBS:	MD = 0.88, Z = 1.8935, Signrank = 117, p = 0.029144 Cohen's d = 0.47	
Phon iTBS:	M = -0.299, rank = 94196 p = 0.942 Cohen's d = -0.358	
Phon cTBS:	M = -1.205, rank = 1 p = 2e-05 Cohen's d = -1.82	
Phon iTBS - cTBS:	MD = 0.91, Z = 3.6773, Signrank = 204, p = 0.00011787 Cohen's d = 0.95	
Tone iTBS:	M = -0.508, rank = 98021 p = 0.98 Cohen's d = -0.523	
Tone cTBS:	M = 0.112, rank = 64266 p = 0.643 Cohen's d = 0.084	
Tone iTBS - cTBS:	MD = -0.62, Z = -1.4201, Signrank = 47, p = 0.92222 Cohen's d = -0.39	
Phon iTBS:	M = -0.045, rank = 68379 p = 0.684 Cohen's d = -0.106	
Phon cTBS:	M = -1.597, rank = 1 p = 2e-05 Cohen's d = -2	
Phon iTBS - cTBS:	MD = 1.55, Z = 3.9013, Signrank = 210, p = 4.7846e-05 Cohen's d = 1.89	

The following values are p-values after FDR correction.

Note that the output for q_fdr_output is in a different order than statistic results. The corresponding conditions are shown below (red color with asterisk for significant results):

q_fdr_output	Condition
0.4750	Left Hemisphere, no noise, iTBS stim., tone
0.9890	Left Hemisphere, no noise, cTBS stim., tone
0.9890	Left Hemisphere, no noise, iTBS stim., consonant
0.0462*	Left Hemisphere, no noise, cTBS stim., consonant
0.7498	Right Hemisphere, no noise, iTBS stim., tone
0.9890	Right Hemisphere, no noise, cTBS stim., tone
0.9890	Right Hemisphere, no noise, iTBS stim., consonant
0.5441	Right Hemisphere, no noise, cTBS stim., consonant
0.7498	Left Hemisphere, noise, iTBS stim., tone
0.0270*	Left Hemisphere, noise, cTBS stim., tone
0.9890	Left Hemisphere, noise, iTBS stim., consonant
0.0002*	Left Hemisphere, noise, cTBS stim., consonant
0.9890	Right Hemisphere, noise, iTBS stim., tone
0.9890	Right Hemisphere, noise, cTBS stim., tone
0.9890	Right Hemisphere, noise, iTBS stim., consonant
0.0002*	Right Hemisphere, noise, cTBS stim., consonant
0.9890	Left Hemisphere, no noise, iTBS-cTBS stim., tone
0.2084	Left Hemisphere, no noise, iTBS-cTBS stim., consonant
0.9890	Right Hemisphere, no noise, iTBS-cTBS stim., tone
0.7658	Right Hemisphere, no noise, iTBS-cTBS stim., consonant
0.0999	Left Hemisphere, noise, iTBS-cTBS stim., tone
0.0007*	Left Hemisphere, noise, iTBS-cTBS stim., consonant
0.9890	Right Hemisphere, noise, iTBS-cTBS stim., tone

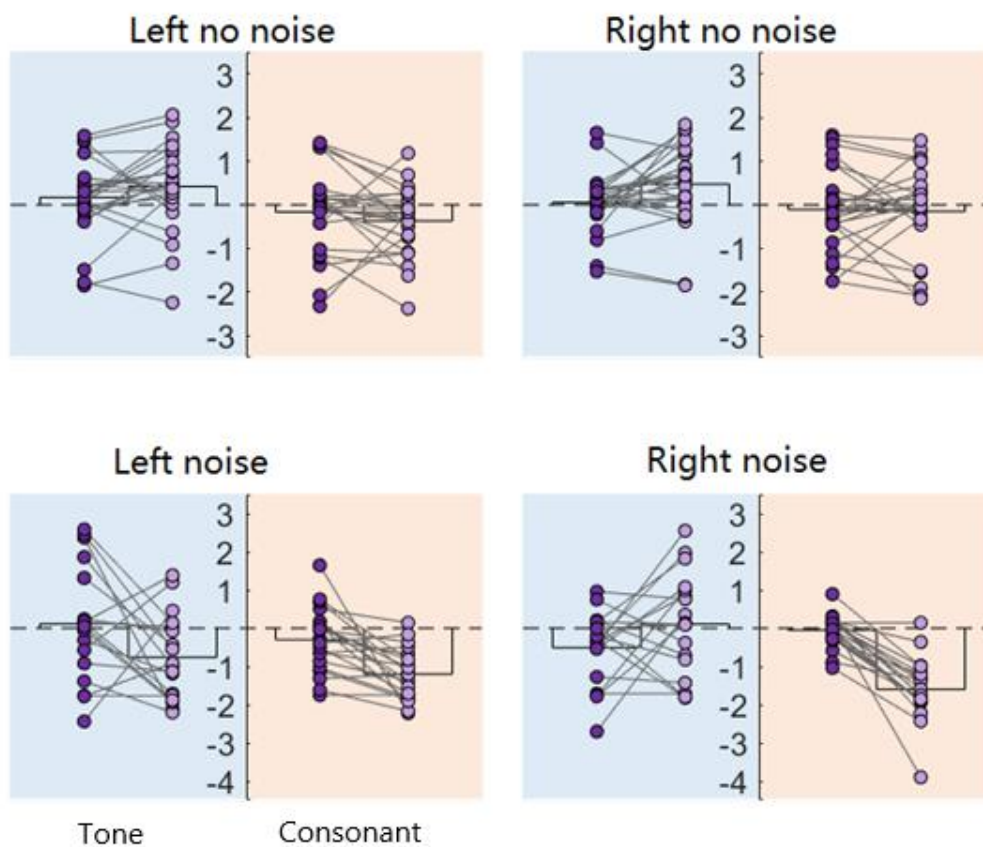
0.0004*

Right Hemisphere, noise, iTBS-cTBS stim., consonant

P-values may slightly be different at each time you run the scripts. The reason is that we use permutation tests where permutation is random. However, after many trials, we confirmed that variations of p-values do not influence levels of significance and our conclusions.

Step 6. Plots are shown directly:

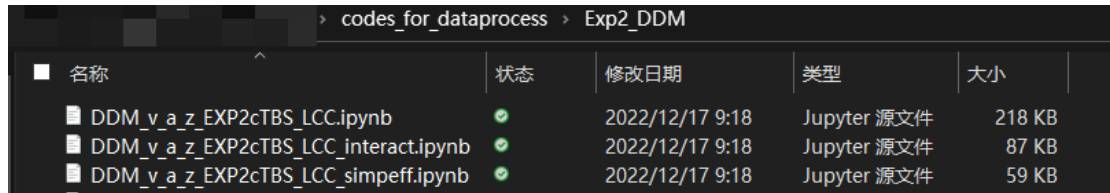
The raw plots were displayed without texts. Below shows the order of plots.



(Dark purple: iTBS; light purple: cTBS)

Experiment 2: drift diffusion model (.\\Exp2_DDM)

Drift diffusion models (DDM) were built independently for each TBS condition (e.g., cTBS upon left dLMC, tone perception in quiet).



The screenshot shows a file explorer window with the path 'codes_for_dataprocess > Exp2_DDM'. It displays a table of files:

名称	状态	修改日期	类型	大小
DDM_v_a_z_EXP2cTBS_LCC.ipynb	✓	2022/12/17 9:18	Jupyter 源文件	218 KB
DDM_v_a_z_EXP2cTBS_LCC_interact.ipynb	✓	2022/12/17 9:18	Jupyter 源文件	87 KB
DDM_v_a_z_EXP2cTBS_LCC_simpeff.ipynb	✓	2022/12/17 9:18	Jupyter 源文件	59 KB

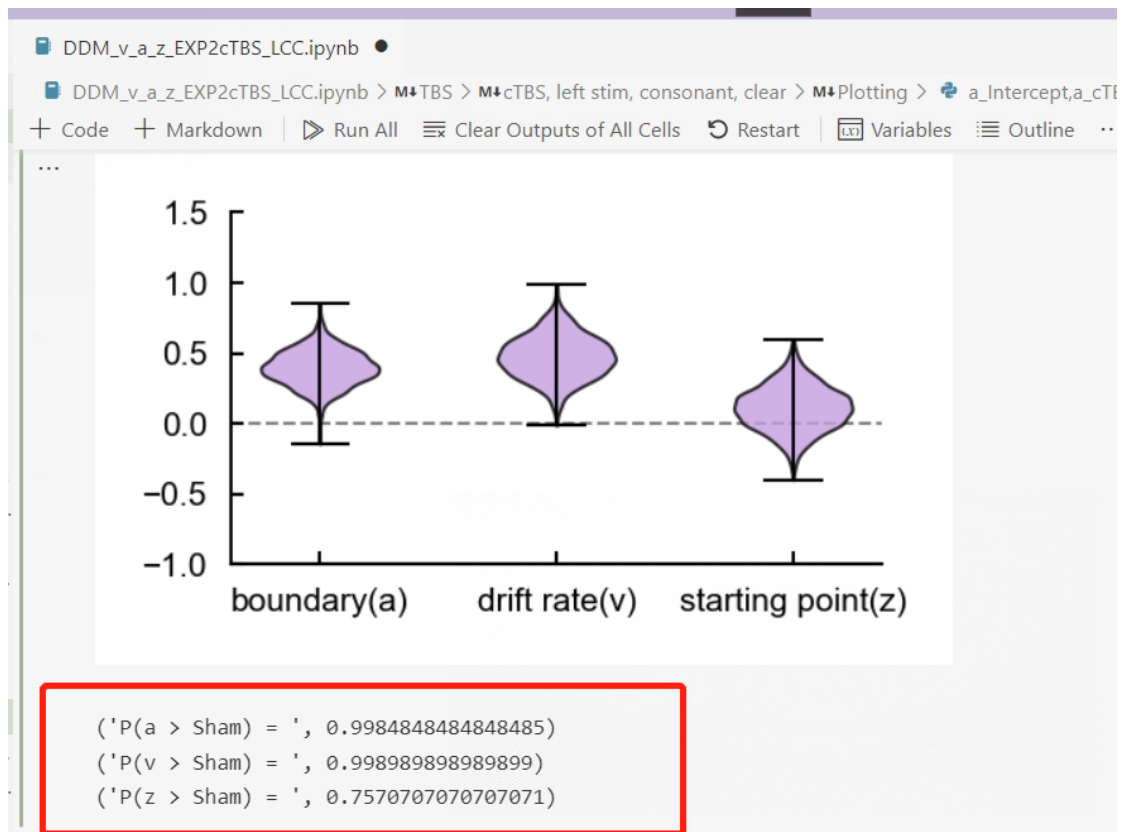
For each condition, three Jupyter notebooks were developed for cTBS effects on DDM parameters (*.ipynb), interactions between cTBS effects and stimuli ambiguity (*_interact.ipynb), simple main effect analyses for how cTBS effects on DDM parameters are modulated by stimuli ambiguity.

(We used Jupyter installed in VScode, and used Anaconda to set up the Python environment. Though later versions may also work, we used Python 2.7.)

All notebooks read the file “exp2_DDM.csv” that was generated by curve fitting codes ('Exp2_slope\data').

Further illustrations can be seen inside notebooks.

Note that the posterior distributions of DDM parameters are plotted, and below the plot are the probabilities that the parameters are over baseline (sham).



Transform these probabilities to two-tailed p-values:

If parameter – sham > 0.5, $p = (1-p)*2$

Else if parameter – sham < 0.5, $p = p*2$

Then do the FDR multiple comparison correction.

See the document for details: **Exp2_DDM\cTBS parameters statistics.docx**.

Experiment 2: reaction time analyses (.\\Exp2_RT)

First, get the reaction time data from each condition and transform them into a readable format for Rstudio (use Jupyter notebook to run this script).

Exp2_RT\\Output_toLMEanalysis_EXP2.ipynb

What the notebook generates are a list of *.csv files:



Then, use R-studio to run this script:

Exp2_RT\\bsliang_lme_cTBSRT_EXP2.R

The output of the script looks like this:

```

> D <- read.csv("Exp2_cTBS_D_L_C_Q_rawdata.csv")
> D$turn_type<-factor(D$turn_type,levels=c("Sham","Left cTBS"))
> D_m <- lmer(rt ~ 1 + turn_type + (1|subj_idx), data=D)
> summary(D_m)
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: rt ~ 1 + turn_type + (1 | subj_idx)
Data: D

REML criterion at convergence: -2069.5

Scaled residuals:
    Min       1Q   Median       3Q      Max
-3.5251 -0.5258 -0.1656  0.2837 14.9033

Random effects:
 Groups   Name      Variance Std.Dev.
subj_idx (Intercept) 0.01299  0.1140
Residual              0.04419  0.2102
Number of obs: 7800, groups:  subj_idx, 25

Fixed effects:
              Estimate Std. Error      df t value Pr(>|t|)
(Intercept)    6.173e-01  2.305e-02 2.452e+01  26.784  <2e-16 ***
turn_typeLeft cTBS 1.163e-02  4.760e-03 7.774e+03   2.443   0.0146 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
      (Intr)
trn_tyLcTBS -0.103

```

Here we focus on the cTBS (turn_type) effects on random intercepts of the model, and get the p-value for each condition and do FDR correction.

See the document for details: **Exp2_RT\ Exp2 iTBS.docx & Exp2 cTBS.docx**