

Supplementary Information

HDDM modelling details -

This analysis requires installation of HDDM. See Wiecki et al., 2013, Front. NeuroInform for more details.

Current link to software download: http://ski.clps.brown.edu/hddm_docs/

```
# import files
import pandas as pd
import matplotlib
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import hddm
# For regressor models
from patsy import dmatrix

# load csv file with single trial data
# for stimulus locked data
data = hddm.load_csv('essp_hddm.csv')
# Check data, e.g.
data.head(10)
```

```
data = hddm.utils.flip_errors(data)
```

#Models

Stimulus varying models

```
m_stim_1 = hddm.HDDM(data[data.response == 1], p_outlier=0.05, depends_on={'a': 'stim', 'v': 'stim'})
# find a good starting point which helps with the convergence.
m_stim_1.find_starting_values()
```

```
m_stim_2 = hddm.HDDM(data[data.response == 1], p_outlier=0.05, depends_on={'v': 'stim', 't': 'stim'})
```

```
m_stim_3 = hddm.HDDM(data[data.response == 1], p_outlier=0.05, depends_on={'a': 'stim', 't': 'stim'})
```

Vary all DDM parameters

```
m_stim_all = hddm.HDDM(data[data.response == 1], p_outlier=0.05, depends_on={'a': 'stim', 'v': 'stim', 't': 'stim'})
```

Within-subject regression model for behavior data

```
m_within = hddm.HDDMRegressor(data[data.response == 1], {"a ~ C(stim, Treatment('NS'))", "v ~ C(stim, Treatment('NS'))", "t ~ C(stim, Treatment('NS'))"}, p_outlier=0.05)
```

For regression models with neural data

Model with emotions modelled together

```

m_reg_conf_cor = hddm.HDDMRegressor(data[data.response == 1],{"a ~ ersp_theta + C(conf, Treatment('LC')) +
ersp_theta:C(conf, Treatment('LC')) + ersp_beta + ersp_beta:C(conf, Treatment('LC'))", "v ~ ersp_theta + C(conf,
Treatment('LC')) + ersp_theta:C(conf, Treatment('LC')) + ersp_beta + ersp_beta:C(conf, Treatment('LC'))"},
p_outlier=0.05)

# Model with varying emotions
m_reg_stim_cor = hddm.HDDMRegressor(data[data.response == 1],{"a ~ ersp_theta + C(stim, Treatment('NS')) +
ersp_theta:C(stim, Treatment('NS')) + ersp_beta + ersp_beta:C(stim, Treatment('NS'))", "v ~ ersp_theta + C(stim,
Treatment('NS')) + ersp_theta:C(stim, Treatment('NS')) + ersp_beta + ersp_beta:C(stim, Treatment('NS'))"},
p_outlier=0.05)

# For running the models
m.find_starting_values()
m.sample(10000, burn=2000) # same for other models

-----

# Check results

# Visualize thresholds and drift rates for different conditions (have to be adjusted to actual parameters)

# e.g. for m_reg_stim_cor:
theta_HS = m_reg_stim_cor.nodes_db.node["a_ersp_theta:C(stim, Treatment('NS'))[T.HS]"]
theta_DS = m_reg_stim_cor.nodes_db.node["a_ersp_theta:C(stim, Treatment('NS'))[T.DS]"]

# Plot the parameters

# e.g. for m_reg_stim_cor:
hddm.analyze.plot_posterior_nodes([theta_DS,theta_HS], bins=20)

# for saving plots, e.g.
hddm.analyze.plot_posterior_nodes([theta_DS,theta_HS], bins=20)
plt.savefig('a_ersp_theta.eps')

# Compare the parameters statistically
# e.g. for m_reg_stim_cor
print "P(Happy > Neutral) = ", (theta_HS.trace() > 0).mean()
print "P(Disgust > Neutral) = ", (theta_DS.trace() > 0).mean()

-----

# Model checks, e.g. for m

m.plot_posteriors()

models = []
for i in range(5):
    m = hddm.HDDM(data)
    m.find_starting_values()
    m.sample(10000, burn=2000)
    models.append(m)
hddm.analyze.gelman_rubin(models)

```

Figures for demonstrating model fitting. All the figures below are plotting using the stimulus – varying model with all the DDM parameters being varied simultaneously.

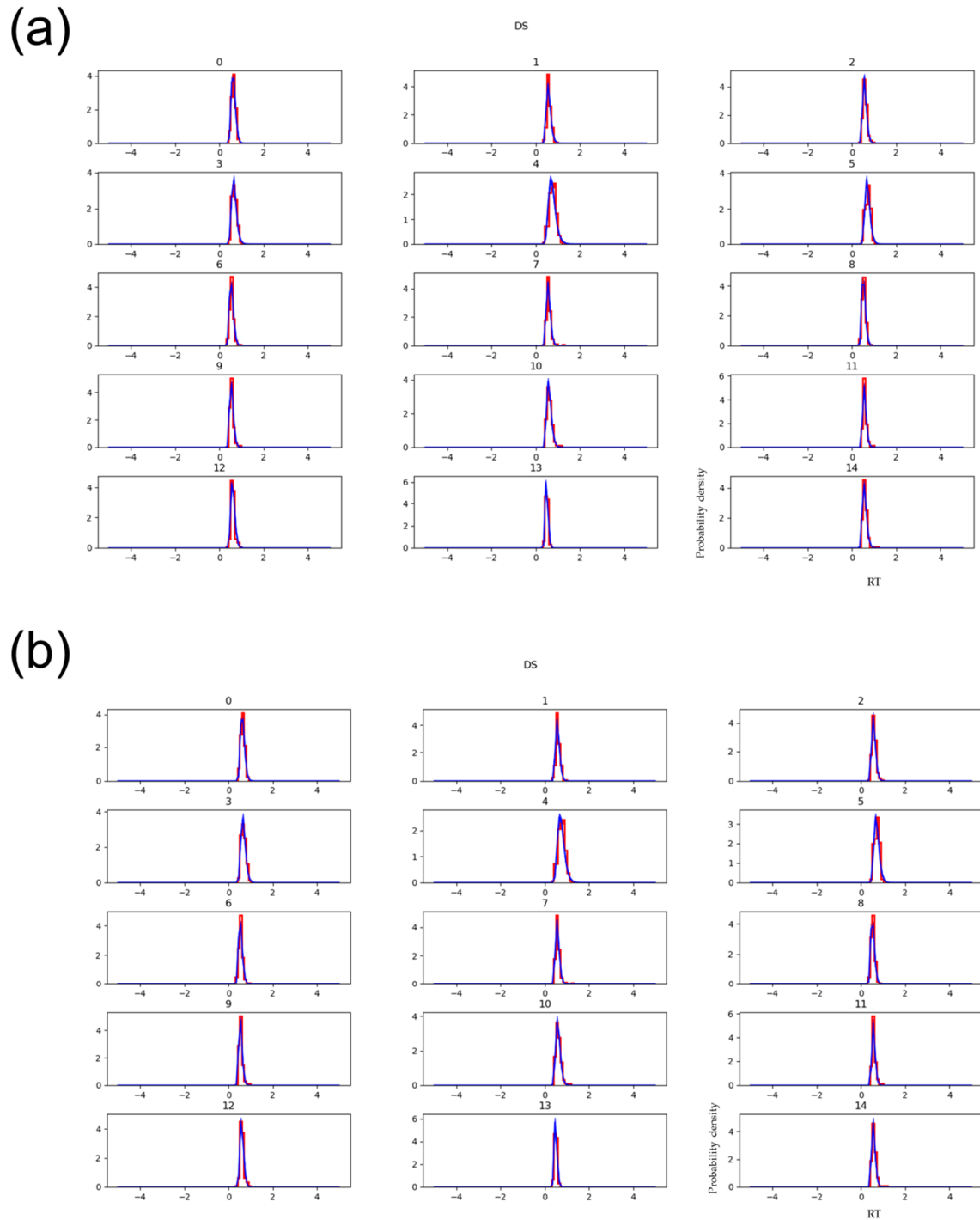


Figure S1: Model fitting accessed by plotting probability density distributions of condition-specific RT data for disgust stimulus (DS): (a) data assumed to having no outliers (b) data assumed to having 5% outliers. The blue lines in (a) and (b) are the

posterior predictives derived from the model while the red lines reflect the RT data. The overlap suggests a good model fit. We have plotted correct responses only since only these values are used in regression models for comparison.

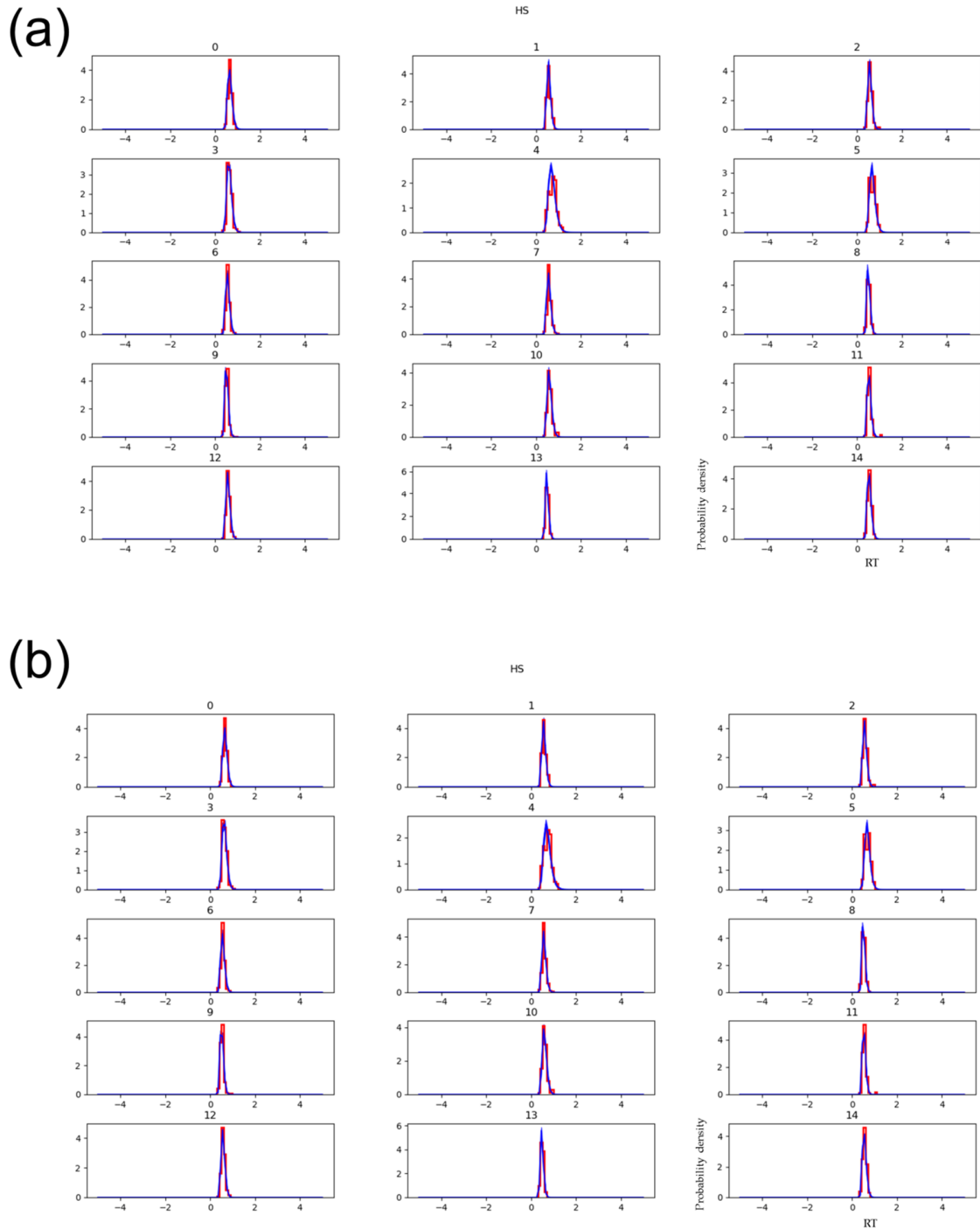


Figure S2: Model fitting accessed by plotting probability density distributions of condition-specific RT data for happy stimulus (HS): (a) data assumed to having no outliers (b) data assumed to having 5% outliers. The blue lines in (a) and (b) are the posterior predictives derived from the model while the red lines reflect the RT data. The

overlap suggests a good model fit. We have plotted correct responses only since only these values are used in regression models for comparison.

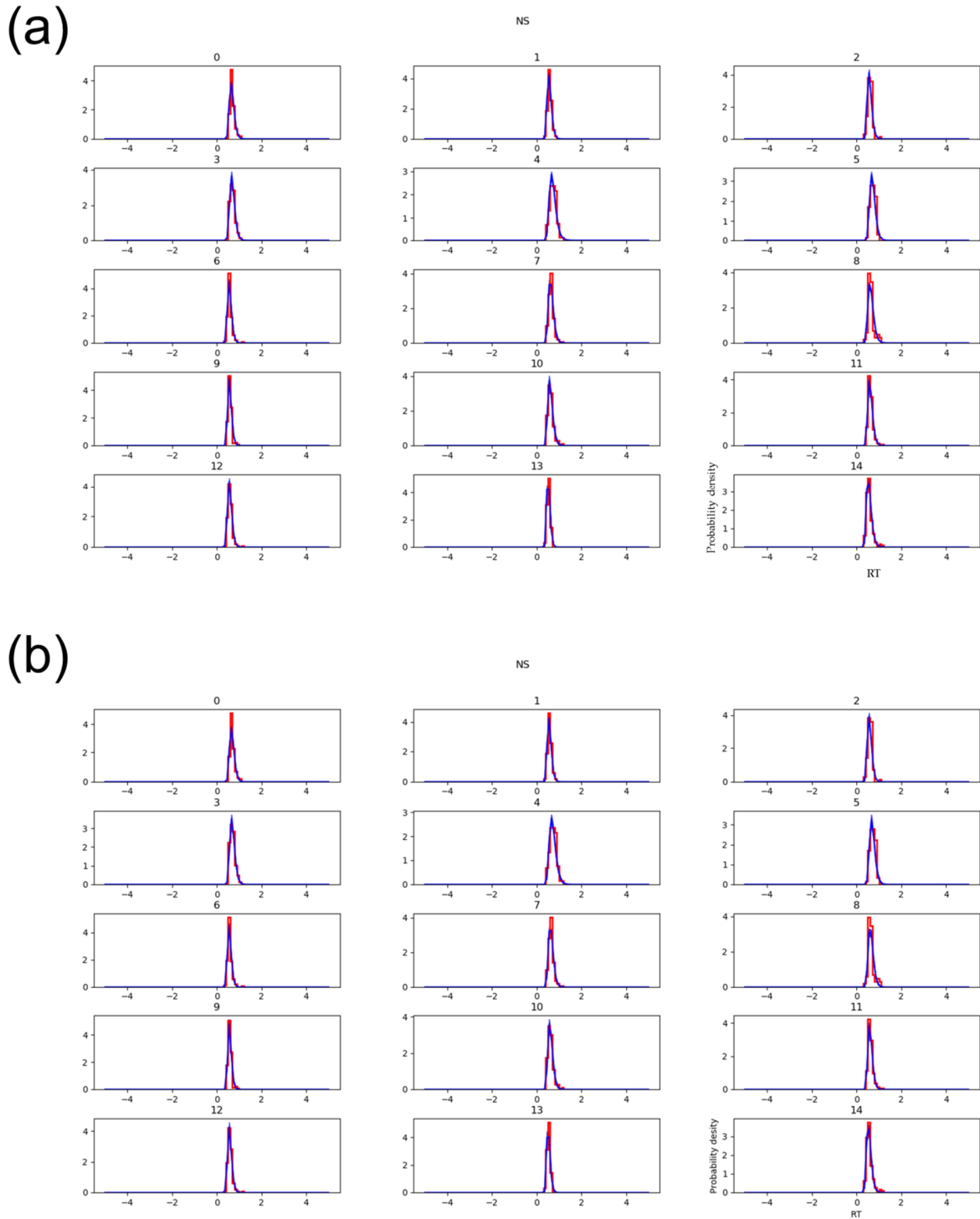


Figure S3: Model fitting assessed by plotting probability density distributions of condition-specific RT data for neutral stimulus (NS): (a) data assumed to having no outliers (b) data assumed to having 5% outliers. The blue lines in (a) and (b) are the posterior predictives derived from the model while the red lines reflect the RT data. The

overlap suggests a good model fit. We have plotted correct responses only since only these values are used in regression models for comparison.